

**Diplomarbeit R-1106-D**

# **Missionsanalyse für den Nanosatelliten CLAVIS**

Kai-Sören Klemich

April 2011

Seiten:	261
Abbildungen:	84
Tabellen:	24

Prof. Dr.-Ing. Peter Vörsmann  
Technische Universität Braunschweig  
Institut für Luft- und Raumfahrtssysteme  
Hermann-Blenk-Straße 23, 38108 Braunschweig

## **Eidesstattliche Erklärung**

Ich erkläre hiermit an Eides Statt, dass ich die nachfolgende Arbeit selbständig und nur unter Zuhilfenahme der angegebenen Literatur angefertigt habe.

---

Unterschrift

Braunschweig, den 19. April 2011

# Übersicht

In der vorliegenden Diplomarbeit werden für den am Institut für Raumfahrtssysteme des DLR entwickelten Nanosatelliten CLAVIS die durch die Solarpaneele erzeugte elektrische Energie, mögliche Bodenkontaktzeiten und die Dauer der Sonnenlichtperioden für verschiedene Orbits mit Hilfe von STK analysiert. Dabei wird zum einen die AISat-Mission betrachtet, für die CLAVIS an eine Nutzlast angepasst wird, die AIS-Signale von Schiffen empfangen soll. Diese Mission soll im August 2011 in einen SSO in 650 km Höhe bei einer LTDN von 10 h gebracht werden und ungefähr ein Jahr dauern. Darüber hinaus werden für zukünftige Missionen verschiedene CLAVIS-Konfigurationen auf beliebigen erdnahen Orbits bis zu einer Bahnhöhe von 650 km betrachtet.

Zu diesem Zweck wird zunächst der theoretische Hintergrund zu Einflüssen auf Eklipsenzeiten auf erdnahen Orbits erarbeitet und anschließend die hier relevanten Subsysteme (Energieversorgungssystem, Kommunikationssystem, Lageregelungssystem) von CLAVIS bzw. AISat erläutert. Zur Simulation der verschiedenen Missionen werden MATLAB-Skripte verwendet, mit denen STK-Szenarios erstellt und die gewünschten Daten gewonnen und analysiert werden.

Für die AISat-Mission werden neben der von den Solarpaneelen erzeugten Energie und der Bodenkontaktzeiten über die gesamte Missionszeit auch das Träger-zu-Rauschverhältnis der Funkverbindung sowie die Parameter der Batterie betrachtet. Für die zukünftigen CLAVIS-Missionen werden Anhaltswerte der Dauer der Sonnenlichtperioden, der erzeugten Energie als Funktion der Dauer einer Sonnenlichtperiode sowie der erzeugten Energie als Funktion der Bahnparameter für verschiedene Konfigurationen und Orbits gegeben. Dabei werden Verläufe aufgezeigt und entsprechende Näherungsgleichungen angegeben, mit denen eine Vorauslegung der entsprechenden Systeme vorgenommen werden kann. Darüber hinaus wird analysiert, welche Konfigurationen auf welchen Orbits in Hinblick auf die zur Verfügung stehende Energie zu bevorzugen sind.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>1</b>
<b>Nomenklatur</b>	<b>5</b>
<b>1 Einleitung</b>	<b>9</b>
<b>2 CLAVIS und AISat</b>	<b>11</b>
2.1 Übersicht . . . . .	11
2.2 Konfigurationen . . . . .	12
2.3 AISat-Mission . . . . .	12
<b>3 Eklipsenzeiten auf erdnahen Orbits</b>	<b>15</b>
3.1 Bedeutung der Eklipsenzeiten . . . . .	15
3.2 Geometrische Betrachtungen . . . . .	16
3.3 Einflüsse auf den Betawinkel und die Eklipsenzeiten . . . . .	20
3.3.1 Einleitende Betrachtungen . . . . .	20
3.3.2 Bahnstörungen . . . . .	21
3.3.3 Geometrie der Erdbahn . . . . .	22
3.3.4 Verlauf des Betawinkels und der Eklipsenzeiten für beliebige Orbits . . . . .	24
<b>4 Energieversorgungssystem</b>	<b>31</b>
4.1 Gesamtsystem . . . . .	31
4.2 Energiebudget . . . . .	34
4.3 Solarpaneele . . . . .	36
4.3.1 Einflüsse der Sonneneinstrahlung . . . . .	36
4.3.2 Degradation . . . . .	37
4.3.3 Einflüsse der Temperatur . . . . .	38
4.3.4 Sonstige Einflüsse . . . . .	39
4.4 Auf- und Entladung der Batterie . . . . .	39
<b>5 Kommunikationssystem</b>	<b>43</b>
5.1 Gesamtsystem . . . . .	43
5.2 Geometrie der Funkverbindung . . . . .	44
5.3 Qualität der Funkverbindung . . . . .	46
5.3.1 Grundlegende Betrachtungen . . . . .	46
5.3.2 Gewinne und Verluste der Funkverbindung . . . . .	47
5.3.3 Rauschen . . . . .	49



<b>6</b>	<b>Lageregelungssystem</b>	<b>51</b>
6.1	Gesamtsystem . . . . .	51
6.2	Gravitationsgradientenstabilisierung . . . . .	52
6.3	Lageverhalten von CLAVIS . . . . .	53
<b>7</b>	<b>Verwendung der Software</b>	<b>54</b>
7.1	SATELLITE TOOL KIT . . . . .	54
7.1.1	Allgemeines . . . . .	54
7.1.2	Satellitenobjekt . . . . .	55
7.1.3	Bodenstationsobjekt . . . . .	57
7.1.4	Sender- und Empfängerobjekt . . . . .	58
7.2	Schnittstelle STK-MATLAB . . . . .	58
<b>8</b>	<b>Durchführung der Simulationen</b>	<b>59</b>
8.1	Übersicht . . . . .	59
8.1.1	AI Sat-Mission . . . . .	59
8.1.2	Zukünftige CLAVIS-Missionen . . . . .	60
8.2	Getroffene Annahmen . . . . .	62
8.2.1	Rotation . . . . .	62
8.2.2	Roll- und Nickschwingungen . . . . .	63
8.2.3	Verwendung des HPOP . . . . .	64
8.3	Berechnung der erzeugten Energie . . . . .	65
8.3.1	Energieberechnungen über das Solar Panel Tool . . . . .	65
8.3.2	Analytische Berechnungen der erzeugten Energie . . . . .	67
8.3.3	Vergleich der beiden Methoden . . . . .	68
<b>9</b>	<b>Ergebnisse für die AI Sat-Mission</b>	<b>70</b>
9.1	Verläufe über einen Orbit . . . . .	70
9.1.1	Verlauf der Einstrahlwinkel . . . . .	70
9.1.2	Verlauf der erzeugten Leistung . . . . .	72
9.1.3	Verlauf der Ladung der Batterie . . . . .	73
9.1.4	Vergleich der verschiedenen Konfigurationen . . . . .	74
9.2	Analyse der Gesamtmission bezüglich der Energieversorgung . . . . .	76
9.2.1	Grundlegende Betrachtungen . . . . .	76
9.2.2	Einfluss der Taumelbewegungen . . . . .	79
9.2.3	Parameter der Batterie . . . . .	81
9.2.4	Zenitausrichtung des Satelliten . . . . .	84
9.3	Bodenkontakt während der AI Sat-Mission . . . . .	86
<b>10</b>	<b>Ergebnisse für zukünftige CLAVIS-Missionen</b>	<b>89</b>
10.1	Näherungsgleichungen . . . . .	89
10.2	Sonnenlichtzeiten . . . . .	90
10.2.1	Darstellung der Ergebnisse . . . . .	90
10.2.2	Orbits mit Inklinationen zwischen 0° und 90° . . . . .	92
10.2.3	Sonnensynchrone Orbits . . . . .	96
10.3	Erzeugte Energie als Funktion der Sonnenlichtdauer . . . . .	100
10.3.1	Grundlegende Betrachtungen . . . . .	100

10.3.2	Gestalt der Näherungsgleichungen . . . . .	101
10.3.3	Standardkonfiguration . . . . .	102
10.3.4	Konfigurationen mit ausklappbaren Paneelen . . . . .	106
10.4	Anhaltswerte der erzeugten Energie als Funktion der Bahnelemente . .	109
10.4.1	Standardkonfiguration . . . . .	109
10.4.2	Konfigurationen mit ausklappbaren Paneelen . . . . .	117
10.5	Aufnehmbare Leistung als Funktion der erzeugten Energie . . . . .	127
10.6	Bodenstationskontakt . . . . .	129
10.7	Evolution der Bahnhöhe über ein Jahr . . . . .	132
10.8	Fazit . . . . .	133
<b>11</b>	<b>Zusammenfassung</b>	<b>136</b>
11.1	Ausblick . . . . .	138
	<b>Literatur</b>	<b>140</b>
	<b>Abbildungsverzeichnis</b>	<b>143</b>
	<b>Tabellenverzeichnis</b>	<b>149</b>
<b>A</b>	<b>Grundlegende MATLAB-Skripte</b>	<b>151</b>
A.1	Häufig verwendete MATLAB-Funktionen . . . . .	151
A.2	MATLAB-Skripte zu den Abbildungen in Kapitel 3 . . . . .	154
<b>B</b>	<b>SPENVIS Eingaben</b>	<b>158</b>
<b>C</b>	<b>Steuerung von STK mit MATLAB</b>	<b>160</b>
C.1	Herstellen der Verbindung zwischen STK und MATLAB . . . . .	160
C.2	Erstellen des STK-Szenarios mit MATLAB . . . . .	161
C.2.1	Definition der Maßeinheiten . . . . .	162
C.2.2	Definition des Szenarioobjekts . . . . .	162
C.2.3	Manipulation der Fenster . . . . .	163
C.2.4	Definition des Satellitenobjekts . . . . .	163
C.2.5	Definition des Bodenstationsobjekts . . . . .	165
C.2.6	Definition des Senderobjekts . . . . .	165
C.2.7	Definition des Empfängerobjekts . . . . .	166
C.3	Erstellen von STK-Reports mit MATLAB . . . . .	167
C.3.1	Sonnenlichtzeiten und Energieerzeugung . . . . .	167
C.3.2	Bodenkontaktzeiten und Übertragungsqualität . . . . .	168
C.3.3	Zugriff auf die Reportdaten . . . . .	169
C.4	Verwendung des HPOP . . . . .	171
<b>D</b>	<b>Grundlegende Skripte für die Simulationen</b>	<b>172</b>
D.1	*.mdl-Dateien . . . . .	172
D.2	attitude_file_generator.m . . . . .	184
D.3	Skript zum Vergleich der Berechnungsmethoden der elektrischen Leistung	185

<b>E</b>	<b>Simulation der AISat-Mission</b>	<b>191</b>
E.1	Skript zur Simulation der AISat-Mission . . . . .	191
E.1.1	Quellcode . . . . .	194
E.2	MATLAB-Skript zur Auswertung der Daten . . . . .	198
E.3	Skripte zur Simulation der Verläufe über einen Orbit . . . . .	202
E.3.1	Skript zur Simulation . . . . .	202
E.3.2	Auswertungsskript . . . . .	205
<b>F</b>	<b>Simulation der zukünftigen CLAVIS-Missionen</b>	<b>207</b>
F.1	Skript zur Simulation zukünftiger CLAVIS-Missionen . . . . .	207
F.1.1	Skript zur Berechnung der Sonnenlichtzeiten und der erzeugten Energie als Funktion der Bahnparameter . . . . .	207
F.1.2	Skript zur Berechnung der erzeugten Energie als Funktion der Sonnenlichtdauer . . . . .	216
F.2	Skripte zur Auswertung der Simulationsergebnisse . . . . .	220
F.2.1	Auswertung der Sonnenlichtzeiten und der erzeugten Energie als Funktion der Bahnparameter . . . . .	220
F.2.2	Auswertung der erzeugten Energie als Funktion der Sonnen- lichtdauer . . . . .	226
F.3	Skript zur Evolution der Bahnhöhe . . . . .	231
<b>G</b>	<b>Zweidimensionale Diagramme</b>	<b>233</b>
<b>H</b>	<b>Übersicht über die zu verwendenden Formeln, Diagramme und Tabellen</b>	<b>242</b>
<b>I</b>	<b>Projektplanung</b>	<b>244</b>
I.1	Work Breakdown Structure . . . . .	244
I.2	Work Package Description . . . . .	245
I.3	Zeitplanung . . . . .	261

# Nomenklatur

## Lateinische Symbole

$A$	Fläche	[m <sup>2</sup> ]
$a$	große Halbachse	[km]
$B$	Bandbreite	[Hz]
$b_{\odot}$	sichtbarer Anteil der Sonne	[-]
$C$	Batteriekapazität	[Wh]
$C^*$	Batteriekapazität	[Ah]
$c$	Lichtgeschwindigkeit im Vakuum, $c = 299.792.458$	[m/s]
$D$	Datenmenge	[MB]
$DoD$	Depth of Discharge	[%]
$d$	Abstand Bodenstation-Satellit	[km]
$E$	Energie	[Wh]
$EIRP$	Equivalent Isotropic Radiated Power	[W]
$e$	Exzentrizität	[-]
$F$	(Verlust-)Faktor	[-]
$f$	Frequenz	[MHz]
$G$	Antennengewinn	[-]
$h$	Kreisbahnhöhe	[km]
$I$	Stromstärke	[A]
$I^*$	Laderate der Batterie als Vielfaches von $C^*$	[C <sup>*</sup> ]
$i$	Inklination	[°]
$i_{\ddot{A}q,Ekl}$	Neigung des Erdäquators gegen die Ekliptik, $i_{\ddot{A}q,Ekl} = 23,5$	[°]
$J$	Massenträgheitsmoment	[kg · m <sup>2</sup> ]
$k$	Boltzmann-Konstante, $k = 1,381 \cdot 10^{-23}$	[J/K]
$L$	Verlustfaktor auf der Funkverbindung	[-]
$LT$	Ortszeit	[h]
$LT DN$	Local Time of Descending Node	[h]
$M$	mittlere Anomalie	[°]
$ML$	mittlere Länge (Bahnelement)	[°]
$m$	Marge zwischen erzeugter und benötigter Energie	[%]
$N$	Rauschleistung	[W]
$n$	Anzahl	[-]
$\vec{n}$	Einheitsnormalenvektor	[-]
$P$	Leistung	[W]
$p$	Leistung pro Flächeneinheit	[W/m <sup>2</sup> ]
$q$	Faktor in Näherungsgleichungen, unterschiedliche Einheiten	
$R_{dat}$	Datenrate	[bit/s]

$R_E$	Erdradius	[km]
$RF$	Battery Return Factor	[-]
$\vec{s}$	Einheitsvektor vom Satelliten zur Sonne	[-]
$T$	Temperatur	[°C]
$t$	Zeit	[s]
$U$	Spannung	[V]
$u$	Bahnwinkel, $u = \varphi + \omega$	[°]
$Z$	Substituent	[-]

## Griechische Symbole

$\alpha$	Öffnungswinkel	[°]
$\beta$	Betawinkel, Winkel zwischen Sonneneinstrahlung und Orbitalebene	[°]
$\gamma$	Abweichung der Einstrahlrichtung von der Flächennormalen	[°]
$\Delta$	Differenz	[-]
$\varepsilon$	Elevation	[°]
$\eta$	Wirkungsgrad	[-]
$\theta$	Winkel zwischen Frühlingspunkt und Sonne in der Ekliptik	[°]
$\kappa$	Winkel zwischen Satellit und Bodenstation am Erdmittelpunkt	[°]
$\lambda$	Wellenlänge	[m]
$\mu$	Gravitationskonstante der Erde, $\mu = 3,986 \cdot 10^{14}$	[m <sup>3</sup> /s <sup>2</sup> ]
$\nu$	Winkel zwischen Nadir und Bodenstation am Satelliten	[°]
$\pi$	Kreiszahl, $\pi = 3,141592654$	[-]
$\varpi$	Länge des Perihels	[°]
$\rho$	halber Winkeldurchmesser der Erde vom Satelliten aus betrachtet	[°]
$\phi$	Winkel zwischen zwei Vektoren	[°]
$\varphi$	wahre Anomalie	[°]
$\Omega$	Rektaszension des aufsteigenden Knotens, RAAN	[°]
$\omega$	Argument des Perizentrums	[°]

## Indizes

0	Ausgangswert unter Standardbedingungen
90	Konfiguration mit um 90° ausgeklappten Paneelen
3dB	3dB-Richtkoppler
acc	Bodenkontakt (access)
act	tatsächlich (actual)
area	Fläche
ass	Baugruppe, Verschaltung (assembly)
atm	Atmosphäre
auf	aufgenommen
ave	durchschnittlich (average)
batt	Batterie
BCR	Battery Charge Receiver
bus	Bus des DEPS
cell	Solarzelle
cha	Aufladung der Batterie (charge)

<i>cov</i>	Abdeckung, Schutzglas (cover)
<i>dB</i>	Angabe in Dezibel
$dB\frac{1}{K}$	Angabe in Dezibel pro Kelvin
<i>dBHz</i>	Angabe in Dezibel Hertz
<i>dBi</i>	Angabe des Antennengewinns in Dezibel bezogen auf einen Isotropstrahler
$dB\frac{J}{K}$	Angabe in Dezibel Joule pro Kelvin
<i>dBK</i>	Angabe in Dezibel Kelvin
<i>dBW</i>	Angabe in Dezibel Watt
<i>deb</i>	Raumfahrtrückstände, Mikrometeoriten (debris)
<i>dis</i>	Entladung der Batterie (discharge)
<i>E</i>	Erde, Erdbahn um die Sonne
<i>ecl</i>	Eklipse, Schattenperiode (eclipse)
<i>eff</i>	effektiv
<i>fs</i>	Freiraumdämpfung (free space loss)
<i>ges</i>	gesamt
<i>grenz</i>	Grenzfall
<i>ill</i>	beleuchtet, Sonnenperiode (illuminated)
<i>incid</i>	Einstrahlungsrichtung (incidence)
<i>init</i>	Initialisierung der Funkverbindung
<i>line</i>	Leitungen
<i>max</i>	maximal
<i>min</i>	minimal
<i>mpp</i>	maximum power point
<i>nom</i>	nominell
<i>oc</i>	open circuit, Leerlauf
<i>pan</i>	Solarpaneel
<i>path</i>	Funkstrecke
<i>point</i>	Ausrichtung der Antenne (pointing)
<i>pol</i>	Polarisation
<i>r</i>	empfangen (received)
<i>rad</i>	Strahlung (radiation)
<i>red</i>	Reduktion
<i>sc</i>	short circuit, Kurzschluss
<i>side</i>	seitliche Solarpaneele
<i>sol</i>	Solarzellen
<i>SSO</i>	Sonnensynchroner Orbit
<i>sys</i>	Gesamtsystem
<i>Tag</i>	ein Tag
<i>t</i>	gesendet (transmitted)
<i>temp</i>	Temperatureinfluss
<i>top</i>	oberes Solarpaneel
<i>U</i>	Umlauf um den Zentralkörper
<i>x</i>	x-Achse
<i>y</i>	y-Achse
<i>z</i>	z-Achse
☉	Sonne
Υ	Frühlingspunkt

## Akronyme

AAU	Aalborg Universitet
AGI	Analytical Graphics Inc.
AIS	Automatic Identification System (Schiffsidentifikationssystem)
AISat	AIS Satellite (Konfiguration von CLAVIS)
AoP	Argument of Pericentre (Argument des Perizentrums, $\omega$ )
AP	Arbeitspaket
BCR	Battery Charge Regulator
BOL	Beginning of Life
BSK	Binary Shift Keying
CAD	Computer Aided Design
CD	Compact Disc
CNR	Carrier-to-Noise-Ratio
DEPS	Deployed Electronic Power System
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DoD	Depth of Discharge
EIRP	Equivalent Isotropic Radiated Power
EOL	End of Life
ESA	European Space Agency
GUI	Graphical User Interface
HPOP	High Precision Orbit Propagator (in STK)
IMU	Inertial Measurement Unit
LEO	Low Earth Orbit
LTAN	Local Time of Ascending Node
LTDN	Local Time of Descending Node
MPP	Maximum Power Point
MPPT	Maximum Power Point Tracking
NaN	Not a Number (ungültiger Wert in MATLAB)
RAAN	Right Ascension of the Ascending Node ( $\Omega$ )
PCM	Power Conditioning Module
PDM	Power Distribution Module
SPENVIS	Space Environment Information System
SSO	Sonnensynchroner Orbit
STK	Satellite Tool Kit (Software von AGI)
TM	Telemetry
UHF	Ultra High Frequency
UTC	Coordinated Universal Time (Koordinierte Weltzeit)
UTCG	Gregorianisches Datum in UTC (in STK)
VVLH	Vehicle Velocity Local Horizontal (Koordinatensystem in STK)

# Kapitel 1

## Einleitung

Am Institut für Raumfahrtssysteme des Deutschen Zentrums für Luft- und Raumfahrt (DLR) in Bremen wird der Nanosatellit CLAVIS entwickelt. Der Name CLAVIS stellt dabei kein Akronym dar, sondern bedeutet in Anlehnung an das Bremer Stadtwappen „Schlüssel“ auf Latein. Ziel der Entwicklung ist ein kompakter Satellit, mit dem kleinere Nutzlasten kostengünstig und schnell zu Testzwecken oder zur Technologiedemonstration in eine Erdumlaufbahn gebracht werden können. Um dies zu erreichen, wird Wert auf ein modulares Design gelegt, bei dem Standardboxen zur Unterbringung der Komponenten verwendet werden. Darüber hinaus sollen sowohl Software als auch elektronische Hardware nach Möglichkeit plug-and-play-fähig sein. Um die Missionskosten so gering wie möglich zu halten, sollen die Bauteile außerdem ohne größeren Aufwand zu fertigen und qualifizieren sein, sofern keine fertig entwickelten Bauteile von Herstellern aus der Industrie bezogen werden können. Mit dem endgültigen Design soll schließlich eine auf Anforderungen der Nutzlasthersteller flexibel einstellbare, missions- und trägerunabhängige Plattform für verschiedenste Nutzlasten bereitgestellt werden.

Die Entwicklung von CLAVIS läuft derzeit in zwei Richtungen. Zum einen wird eine „Standardkonfiguration“ entwickelt, die die oben beschriebenen Eigenschaften haben und auf späteren Missionen für verschiedene Nutzlasten verwendet werden soll. Die Grundversion dieses Satelliten ist annähernd würfelförmig mit einer Kantenlänge von ca. 30cm und besitzt an fünf Außenflächen fest installierte Solarpaneele. Darüber hinaus sind allerdings auch andere Konfigurationen mit weniger Solarpaneelen zur Akkommodation einer Nutzlast oder ausfaltbaren Solarpaneelen in Planung. Zum anderen wird die CLAVIS-Plattform an eine verhältnismäßig große Nutzlast angepasst, die unter dem Namen AISat entwickelt wurde. Ziel dieser Mission, die im August 2011 starten soll, ist es, AIS-Signale (Automatic Identification System) von Schiffen zu empfangen. Das AIS ist ein Funksystem, mit dem größere Schiffe ausgerüstet sind, um Identifikations- und Navigationsdaten auszutauschen und somit die Sicherheit auf See zu erhöhen. Die Reichweite dieses Systems ist jedoch praktisch auf Sichtweite begrenzt. Die AISat-Nutzlast besitzt daher eine speziell entwickelte Antenne mit hohem Antennengewinn, mit der die AIS-Signale empfangen werden können. Damit soll gezeigt werden, dass es möglich ist, die Signale aller Schiffe per Satellit zu empfangen um somit die Einschränkungen der Reichweite des Systems aufzuheben. Da diese Nutzlast jedoch verhältnismäßig groß und schwer ist, muss das



Design der Standardkonfiguration entsprechend angepasst werden.

In dieser Arbeit wird eine Missionsanalyse für CLAVIS durchgeführt. Dazu werden zunächst die Eklipsenzeiten, die erzeugte Energie, die Parameter der Batterie, die Bodenkontaktzeiten und die übertragbare Datenmenge für die AISat-Mission eingehend betrachtet. Für zukünftige CLAVIS-Missionen ist die wichtigste Einschränkung der möglichen Orbits die Bahnhöhe, da CLAVIS in keiner Konfiguration ein aktives Bahnregelungssystem besitzen wird. Um den European Code of Conduct for Space Debris Mitigation [1] zu erfüllen, muss die Bahnhöhe auf ca. 650 km beschränkt werden, so dass der Satellit nach weniger als 25 Jahren wiedereintritt. Darüber hinaus gibt es jedoch keine weiteren Einschränkungen für mögliche CLAVIS-Missionen, weshalb ein möglichst großer Teil aller denkbaren, niedrigeren Orbits betrachtet wird. Die Eklipsenzeiten und die damit verbundenen Änderungen der durch die Solarpaneele erzeugten Energie sowie die Bodenkontaktzeiten werden für verschiedene Orbits bestimmt. Dabei werden die verschiedenen Konfigurationen von CLAVIS einzeln untersucht, da sich die erzeugte Energie aufgrund der unterschiedlichen Anordnungen der Solarpaneele ändert. Die gewünschten Daten werden mit Hilfe der durch MATLAB-Skripte gesteuerten Software SATELLITE TOOL KIT (STK) bestimmt, analysiert und aufbereitet dargestellt.

# Kapitel 2

## CLAVIS und AISat

### 2.1 Übersicht

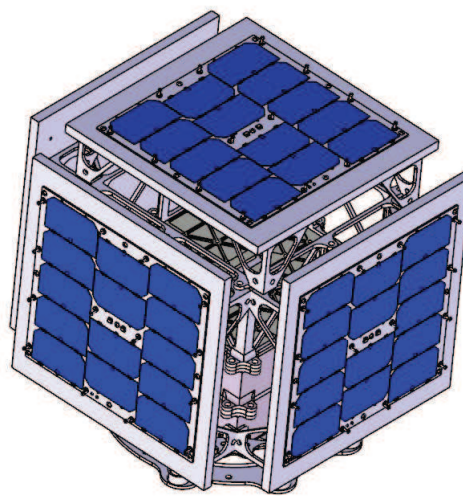
Wie bereits in der Einleitung beschrieben ist CLAVIS ein annähernd würfelförmiger Nanosatellit mit einer Masse von ca. 10 kg und einer Kantenlänge von ca. 30 cm. Zur Erzeugung von elektrischer Energie besitzt CLAVIS in der Standardkonfiguration Solarpaneele an fünf von sechs Außenflächen. Die einzige Seite ohne Solarpaneel wird während des Starts zur Befestigung des Satelliten in der Ladebucht des Trägersystems genutzt. Nachdem der Zielorbit erreicht ist, soll sich an dieser Seite ein Ausleger zur Gravitationsgradientenstabilisierung („Gravity-Gradient-Boom“) entfalten, mit dem eine Ausrichtung des Satelliten in Nadirrichtung erreicht wird. Zur weiteren Stabilisierung der Lage sind Magnet Torquer vorgesehen, die ein geeignetes Magnetfeld aufbauen, das mit dem Erdmagnetfeld interagiert, so dass die Lage des Satelliten besser gehalten werden kann. Zur Speicherung der während der Zeit im Sonnenlicht erzeugten Energie ist ein Lithium-Polymer-Batterieblock vorgesehen. Ein darauf zugeschnittenes Energieverteilungssystem sorgt für die korrekte Einstellung des Batterieladestroms und der Versorgungsströme für die einzelnen Verbraucher.

Zum Empfang der von CLAVIS gesammelten Daten am Boden wird am Institut für Raumfahrtssysteme des DLR in Bremen eine Empfangsstation eingerichtet. Darüber hinaus kann in der Zukunft auch die sich noch im Aufbau befindliche Bodenstation des DLR in Inuvik im Norden Kanadas genutzt werden. Durch ihre polnahe Lage kann von dort aus der Kontakt zu Satelliten auf hochinklinierten Umlaufbahnen über vergleichsweise lange Zeiträume aufrecht erhalten werden. Für die Standardkonfiguration des CLAVIS sind Antennen zur Kommunikation im S-Band und im Amateurfunkband angedacht.

Grundsätzlich sind alle denkbaren erdnahen Orbits unterhalb von ca. 650 km Bahnhöhe für den Betrieb von verschiedensten Nutzlasten auf CLAVIS denkbar. Im Rahmen dieser Arbeit wird der Bereich der Umlaufbahnen mit Inklinationen zwischen  $0^\circ$  und  $90^\circ$  sowie der Sonnensynchronen Orbits (SSO) (siehe Abschnitt 3.3.2) im relevanten Bahnhöhenbereich systematisch betrachtet.

## 2.2 Konfigurationen

Die Standardkonfiguration des CLAVIS ist in Abbildung 2.1 dargestellt, wobei eines der Solarpaneele fehlt. Grundsätzlich ist es natürlich denkbar, weniger als fünf Paneele zu verwenden, um weiteren Platz für eine Nutzlast zu schaffen, sofern genug elektrische Energie zur Verfügung steht. Das grundlegende Design der Standardkonfiguration ist abgeschlossen, jedoch ruht die weitere Entwicklung derzeit weitestgehend, da für die AISat-Mission bereits ein Startdatum feststeht. Um diesen Termin einhalten zu können, konzentrieren sich die Entwicklungen zur Zeit auf diese Konfiguration (siehe Abschnitt 2.3).

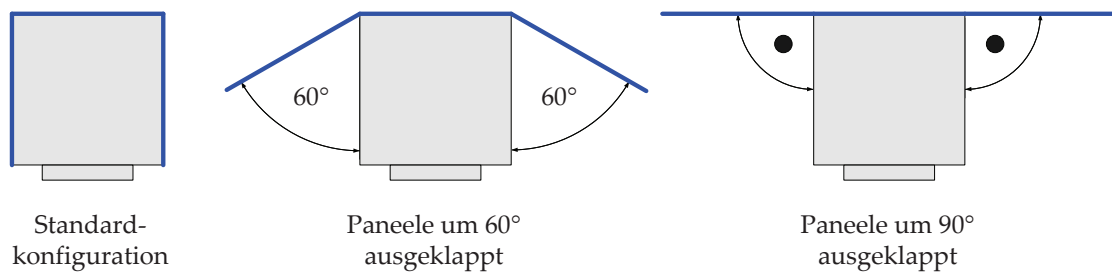


**Abbildung 2.1:** CLAVIS in der Standardkonfiguration

Für eine weitere hier untersuchte Konfiguration wird ein Mechanismus entwickelt, mit dem die Solarpaneele bewegt werden können [2]. Ziel ist es, die vier seitlichen Paneele in Richtung des oberen Panels, das der Seite mit dem Gravity-Gradient-Boom gegenüberliegt, um einen Winkel von  $90^\circ$  auszuklappen. Somit entsteht praktisch eine einzige Fläche, wie in Abbildung 2.2 schematisch dargestellt. Eine solche Konfiguration kann in bestimmten Umlaufbahnen, bei denen der Winkel zwischen Orbitalebene und Sonneneinstrahlung gering ist, deutlich mehr Energie erzeugen als die Standardkonfiguration. Hier wird, wie ebenfalls in der Abbildung dargestellt, zusätzlich eine Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen betrachtet. Dieser Fall kann als Worst-Case-Szenario angesehen werden, für den Fall, dass die Arretierung des Ausklappmechanismus versagt [2]. Im Rahmen der Entwicklung des Ausklappmechanismus wird auch ein anderes Design der Solarpaneele vorgesehen, so dass mehr Solarzellen pro Panel Platz finden.

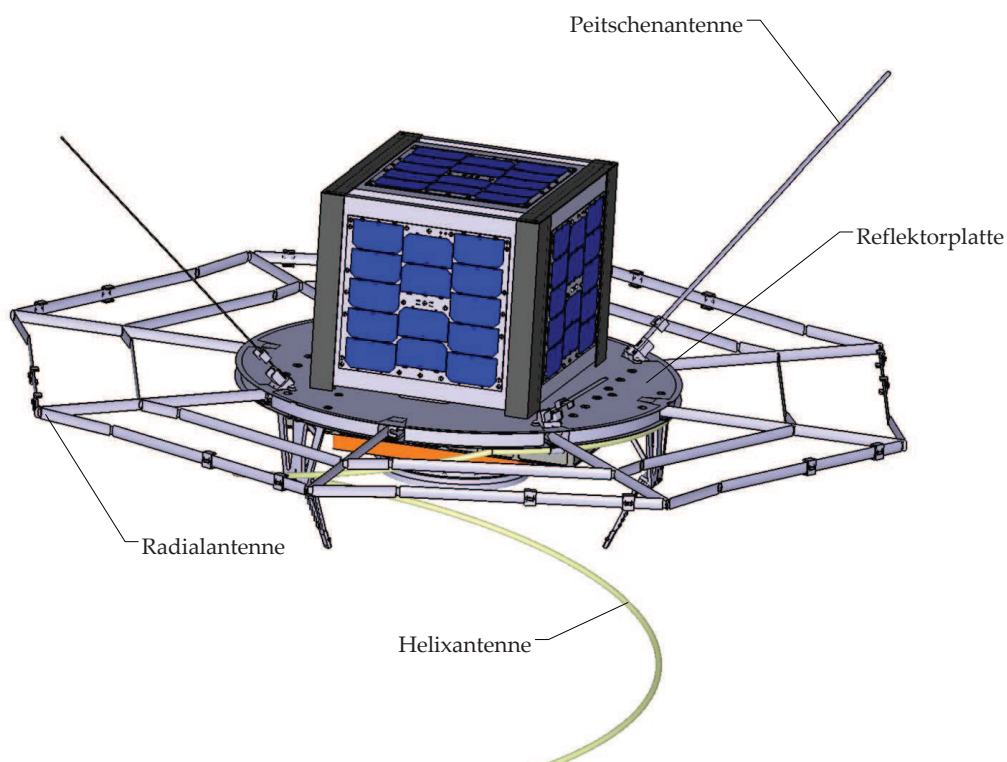
## 2.3 AISat-Mission

Wie bereits erwähnt soll die AISat-Mission als erste CLAVIS-Mission im August 2011 von einer indischen Trägerrakete in einen SSO in einer Höhe von 650 km gebracht wer-



**Abbildung 2.2:** Schematische Darstellung der Konfigurationen mit ausklappbaren Paneelen

den. Die Mission soll ungefähr ein Jahr dauern. Die Definition der Subsysteme und die Implementierung der notwendigen Software sind derzeit in vollem Gange. In dieser Konfiguration wird der Standard-CLAVIS an die schon vorher entwickelte AISat-Nutzlast angepasst. Diese besteht im wesentlichen aus einer Reflektorplatte mit einem Durchmesser von knapp 60 cm, an der eine Helixantenne sowie eine Radialantenne befestigt sind, mit denen die relativ schwachen AIS-Signale empfangen werden können. Mit diesen Antennen soll es außerdem möglich sein, Signale von zwei Schiffen zu trennen, die sich sehr nah beieinander befinden, was bisher aus dem All unmöglich war. Beide Antennen sind während des Starts zusammengestaucht und werden nach dem Einschuss in den Zielorbit durch einen einzigen Mechanismus entfaltet. Die Helixantenne erreicht danach eine Länge von mehreren Metern. Abbildung 2.3 zeigt CLAVIS in der AISat-Konfiguration.



**Abbildung 2.3:** CLAVIS in der AISat-Konfiguration

Man erkennt den oberen Teil der Helixantenne und die Reflektorplatte, auf der der Hauptkörper des Satelliten befestigt ist. In der Ebene der Reflektorplatte ist außerdem die Radialantenne zu sehen. Auf der Reflektorplatte befinden sich darüber hinaus drei Peitschenantennen im  $45^\circ$ -Winkel nach außen geneigt. Zwei von diesen Antennen sind relativ lang und dienen ebenfalls dem Empfang der AIS-Signale. Die kürzere dieser Antennen wird hingegen für die Kommunikation mit der Bodenstation im Amateurfunkband verwendet. Für die AISat-Mission wird lediglich dieser Wellenlängenbereich verwendet und es ist keine Sende- oder Empfangsanlage für das S-Band vorgesehen. Auf der Abbildung nicht sichtbar ist die zweite Peitschenantenne zur Kommunikation mit der Bodenstation. Sie befindet sich in der Mitte der Unterseite der Reflektorplatte und zeigt im nominellen Betrieb Richtung Nadir. Beide dieser Monopolantennen bilden zusammen eine Dipolantenne mit der die gesammelten Daten zur Erde gefunkt werden. Um weiteres Gewicht zu sparen werden auf der AISat-Konfiguration jedoch nicht die in der Abbildung dargestellten Standardsolarpaneele verwendet, sondern etwas kleinere Paneele mit jeweils 12 Solarzellen (siehe auch Abschnitt 4.1).

Im Folgenden werden zunächst die theoretischen Grundlagen für die Bestimmung der Eklipsen- bzw. Sonnenlichtzeiten auf niedrigen Erdorbits behandelt. Eine Beschreibung der für diese Arbeit relevanten Systeme der AISat-Konfiguration schließt sich in den darauf folgenden Kapiteln an.

# Kapitel 3

## Eklipsenzeiten auf erdnahen Orbits

### 3.1 Bedeutung der Eklipsenzeiten

Für die Missionsanalyse jedweder Satellitenmission in den erdnahen Bereich spielt die Kenntnis der Zeiten, die der Satellit im Erdschatten (Eklipse) bzw. im direkten Sonnenlicht verbringt, eine wichtige Rolle. Die Einstrahlung der Sonne hat dabei vor allem einen Einfluss auf die Energieversorgung und den Thermalhaushalt aber auch auf die Lage und die Bahn eines Satelliten. Die einzelnen Einflüsse werden im Folgenden kurz beschrieben.

Für die Energieversorgung praktisch aller Satelliten auf niederen Erdborbits (LEO) mit einer Lebensdauer von mehr als wenigen Tagen oder Wochen werden Solarpaneele verwendet. Diese laden während der Sonnenlichtperioden eine Batterie auf, mit der die Energieversorgung während der Schattenperioden sichergestellt wird. Neben der bloßen akkumulierten Dauer der Sonneneinstrahlung über die Dauer einer Mission ist daher auch der Verlauf der Dauer der Eklipsen- und Sonnenlichtzeiten von Bedeutung, da dieser die Ladezyklen der Batterie bestimmt. Nähere Ausführungen dazu finden sich in Kapitel 4.

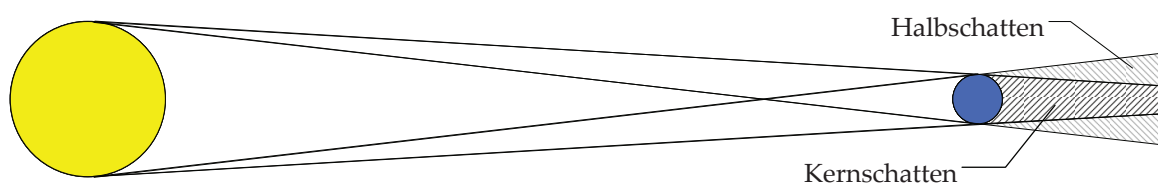
Aufgrund des vorherrschenden Vakuums im Weltraum ist die einzige Möglichkeit zum Wärmeaustausch eines Raumfahrzeugs mit der Umgebung die Wärmestrahlung. Zur Wärmebilanz hinzu kommt lediglich die an Bord aufgewendete elektrische Energie, die zu einem Großteil in Wärmeenergie übergeht. Während sich ein Satellit im Sonnenlicht befindet, nimmt er Strahlungsenergie von der Sonne und, sofern er sich im erdnahen Bereich befindet, auch von der Erde auf. Gleichzeitig strahlt der Satellit über seine Außenflächen auch Wärme ab. Bei Satelliten ohne aufwändiges Thermalkontrollsystem wie CLAVIS ist die Wärmebilanz dabei stets positiv; der Satellit heizt sich auf. Während Eklipsenperioden ist das Gegenteil der Fall. Die einzige nennenswerte Wärmequelle neben der aufgewendeten elektrischen Energie der Satellitensysteme ist dann die Nachtseite der Erde, so dass die Abstrahlung von Wärme ins All überwiegt; der Satellit kühlt sich ab. Aufgrund der normalerweise signifikanten Wärmekapazität und der Trägheit der Wärmeleitung innerhalb eines Satelliten erreicht ein Satellit ohne Thermalkontrolle, der regelmäßig Eklipsen durchläuft, im Normalfall zu keinem Zeitpunkt das thermische Gleichgewicht. Die Temperatur aller Komponenten kann daher innerhalb eines Orbits um mehr als 100 °C

schwanken, was ein wichtiges Kriterium für die Auswahl bzw. die Auslegung der Teile darstellt. Dies gilt insbesondere auch für die Solarpaneele, deren Wirkungsgrad bei höheren Temperaturen abnimmt (siehe Kapitel 4).

Die Strahlung der Sonne hat darüber hinaus auch einen mechanischen Einfluss auf ein Raumfahrzeug. Die Photonen des Sonnenlichts üben auf jeden Körper einen geringen Strahlungsdruck aus, der sowohl die Lage eines Satelliten im Raum als auch den Orbit beeinflussen kann. Außerdem heizt die Sonne die Atmosphäre der Erde auf, was zu einer erhöhten Dichte in hohen Atmosphärenschichten führt. Somit erhöht sich der aerodynamische Widerstand von niedrigfliegende Satelliten auf der Tagseite der Erde im Allgemeinen und am Subsolarpunkt im Besonderen, was letztlich zu einer stärkeren Absenkung des Orbits führt. Aufgrund all dieser Einflüsse der Sonnen- und Schattenzeiten ist die genaue Kenntnis der Länge und der Abfolge dieser Perioden wünschenswert. Im Folgenden werden diese daher näher betrachtet.

## 3.2 Geometrische Betrachtungen

Da die Sonne von der Erde aus betrachtet nicht als punktförmige Lichtquelle angesehen werden kann, gibt es einen Bereich an den Rändern des Erdschattens, in dem die Sonne nur teilweise von der Erde bedeckt ist. Dieser Bereich wird als Halbschatten (Penumbra) bezeichnet. Befindet sich ein Raumfahrzeug in diesem Bereich, empfängt es eine geringere Strahlungsleistung von der Sonne. Diese kann in einer Rechnung mit einem Faktor  $b_{\odot}$  berücksichtigt werden, der dem Anteil der sichtbaren Sonnenfläche an der Gesamtsonnenfläche entspricht. Während einer Eklipsenperiode befindet sich ein Raumfahrzeug zumeist jedoch im Kernschatten (Umbra) der Erde, wo die Sonne komplett von der Erde verdeckt ist. Der Effekt ist in Abbildung 3.1 verdeutlicht.



**Abbildung 3.1:** Kernschatten und Halbschatten der Erde, nicht maßstabsgetreu

Die hier betrachteten Eklipsen entstehen dadurch, dass ein Raumfahrzeug in LEO regelmäßig in den Erdschatten eintritt. Grundsätzlich sind darüber hinaus auch Eklipsen durch andere Raumfahrzeuge oder den Mond denkbar. Aufgrund der geringen Größe künstlicher Satelliten sind erstere jedoch sehr selten und sehr kurz und somit vernachlässigbar. Eklipsen durch den Mond entstehen für erdnahe Satelliten meist in unmittelbarer zeitlicher Nähe zu Sonnenfinsternissen, da dann der Schatten des Mondes in die Nähe der Erde kommt. Dies ist dann der Fall, wenn Erde, Sonne und Mond auf einer Geraden liegen, was aufgrund der Neigung der Mondbahnebene gegen die Ekliptik nur bei jedem fünften oder sechsten Neumond, also im Abstand von ca. einem halben Jahr auftreten kann. Da der Kernschatten des Mondes zudem verhältnismäßig klein ist, befindet sich ein Raumfahrzeug wenn überhaupt, dann nur

für einen sehr kurzen Zeitraum im Kernschatten. Daher werden auch die Eklipsen durch den Mond im Folgenden vernachlässigt.

Bei der Betrachtung von Eklipsen im erdnahen Bereich spielt der Winkel zwischen der Bahnebene eines Raumfahrzeugs und der einfallenden Sonnenstrahlung eine wichtige Rolle. Der kleinere der beiden Winkel zwischen dem Sonnenvektor und der Bahnebene wird als Betawinkel (Beta Angle)  $\beta$  bezeichnet. Dieser Winkel kann Werte zwischen  $-90^\circ$  und  $+90^\circ$  annehmen, wobei er als positiv definiert wird, wenn der Drehimpulsvektor des Orbits des betrachteten Raumfahrzeugs in Richtung der sonnenzugewandten Seite des Orbits zeigt. Anders ausgedrückt ist der Betawinkel positiv, wenn der Satellit von einem Beobachter in der Sonne aus betrachtet entgegen dem Uhrzeigersinn umläuft. Aus der Anschauung ergibt sich, dass die Eklipsendauer minimal wird, wenn der Betrag des Betawinkels maximal wird. Ist  $\beta = \pm 90^\circ$  scheint die Sonne senkrecht zur Orbitalebene, so dass der gesamte Orbit in der Sonne liegt und keine Eklipse auftritt. Ist hingegen  $\beta = 0^\circ$  scheint die Sonne gerade auf die Orbitalebene, so dass der größtmögliche Anteil des Orbits hinter der Erde liegt; die Eklipsendauer ist maximal.

Aus Abbildung 3.1 wird ersichtlich, dass Erdhalb- und -kernschatten eine konische Form aufweisen. In der Abbildung sind die Größen und Entfernungen jedoch nicht maßstabsgetreu wiedergegeben. Berücksichtigt man die im Vergleich zur Entfernung zwischen Erde und Raumfahrzeug um viele Größenordnungen größere Entfernung zwischen Erde und Sonne, so kann man die Strahlen als näherungsweise parallel annehmen, so dass es nur noch einen zylinderförmigen Kernschattenbereich gibt. Dabei werden außerdem die Abweichungen der Erde von der Kugelform vernachlässigt. Mit diesen Vereinfachungen lässt sich die Dauer einer Eklipse bei gegebenem Orbit und gegebenem Zeitpunkt verhältnismäßig einfach analytisch bestimmen, weshalb sie für die folgenden Betrachtungen verwendet wird. Die Betrachtungen sind angelehnt an die Ausführungen in [3].

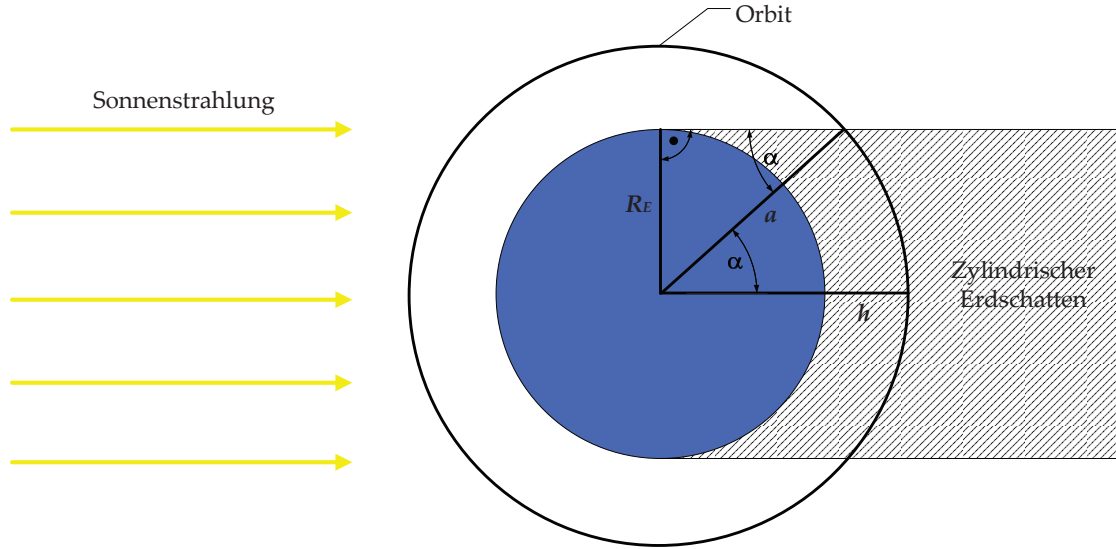
Unter dieser Voraussetzung hängt die maximal mögliche Eklipsendauer auf einem kreisförmigen Orbit nur von der Höhe des Orbits ab und kann leicht bestimmt werden. Abbildung 3.2 zeigt die Geometrie eines Orbits in der Draufsicht auf die Orbitalebene mit dem zylindrischen Schatten der Erde, wobei der Sonnenvektor in der Orbitalebene liegt, so dass  $\beta = 0^\circ$  ist. Darin sind:

$a$	große Halbachse des Orbits
$h$	Bahnhöhe des Orbits
$R_E$	Erdradius
$\alpha$	Öffnungswinkel

Für den Orbit mit der großen Halbachse  $a = R_E + h$  ergibt sich die Dauer der Eklipse  $t_{ecl}$  aus der Umlaufzeit  $t_U$  und dem Öffnungswinkel  $\alpha$ . Dabei ist das Verhältnis von der Eklipsendauer zur Umlaufdauer gleich dem Verhältnis des doppelten Öffnungswinkels zu  $360^\circ$ . Daraus folgt dann die Gleichung für die Eklipsendauer

$$t_{ecl} = t_U \cdot \frac{2\alpha}{360^\circ} \quad (3.1)$$





**Abbildung 3.2:** Geometrie zur näherungsweisen Berechnung der maximalen Eklipsendauer für  $\beta = 0^\circ$

mit der Umlaufdauer

$$t_U = 2\pi \sqrt{\frac{a^3}{\mu}} \quad (3.2)$$

mit  $\mu = 3,986 \cdot 10^{14} \frac{\text{m}^3}{\text{s}^2}$  als der Gravitationskonstanten der Erde. Für den hier dargestellten Fall ist wie gesagt  $\beta = 0^\circ$  und damit  $t_{ecl}$  maximal. Aus Gleichung (3.1) ist ersichtlich, dass damit auch  $\alpha$  maximal sein muss. Aus dem rechtwinkligen Dreieck in Abbildung 3.2 lässt sich nun für den maximalen Öffnungswinkel  $\alpha_{max}$  ablesen

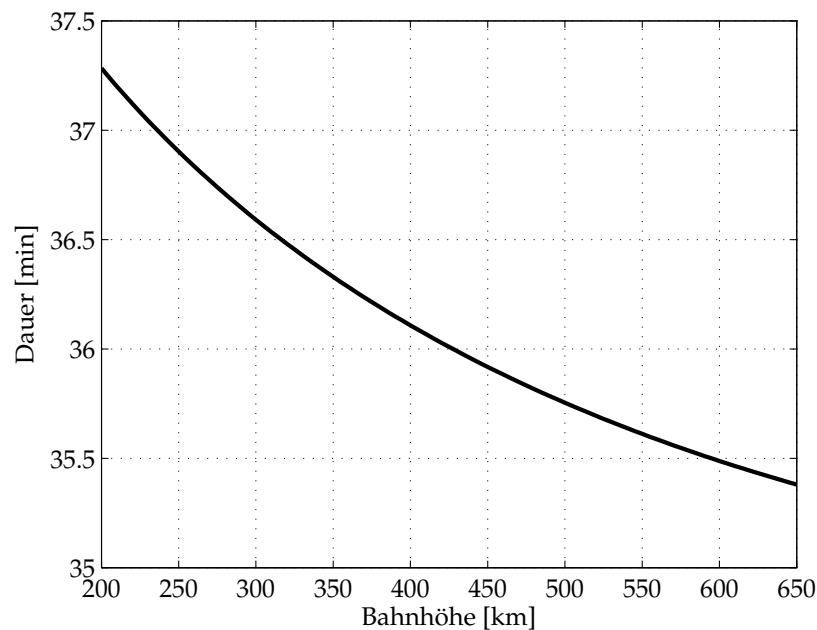
$$\sin(\alpha_{max}) = \frac{R_E}{R_E + h} = \frac{R_E}{a} \quad (3.3)$$

Setzt man nun Gleichung (3.3) in Gleichung (3.1) ein, erhält man eine Bestimmungsgleichung für die maximale Eklipsendauer  $t_{ecl,max}$  in Abhängigkeit der Bahnhöhe:

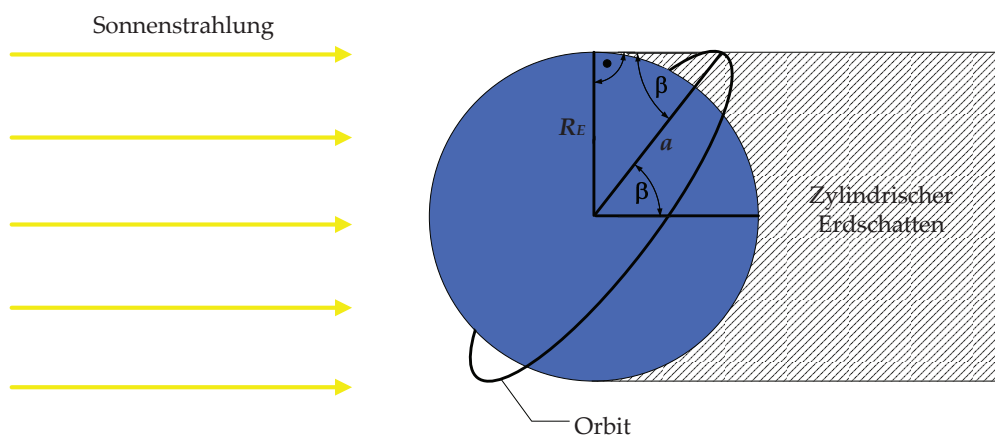
$$t_{ecl,max} = t_U \cdot \frac{\arcsin\left(\frac{R_E}{R_E + h}\right)}{180^\circ} \quad (3.4)$$

Abbildung 3.3 zeigt den Verlauf der maximalen Eklipsendauer über die hier betrachteten Bahnhöhen, Gleichung (3.4) folgend.

Betrachtet man die geometrische Situation für ein variables  $\beta \neq 0^\circ$ , ist die Bestimmung von  $\alpha$  komplizierter. Hierbei reicht die ebene Betrachtung nicht mehr aus, da der Orbit in Abhängigkeit des Betawinkels und der Bahnhöhe nicht mehr den gesamten Durchmesser des Erdschattens durchläuft, sondern nur noch einen kleineren Anteil. Um diesen zu bestimmen sind Betrachtungen in sphärischer Geometrie nötig, weshalb hier auf eine Herleitung verzichtet wird. In Abhängigkeit der Bahnhöhe lässt sich jedoch der Grenzfall bestimmen, bei dem gerade die gesamte Umlaufbahn in der Sonne liegt.



**Abbildung 3.3:** Maximale Eklipsendauer als Funktion der Bahnhöhe für kreisförmige Orbits



**Abbildung 3.4:** Geometrie zur Berechnung des Grenzfalles des Betawinkels, bei dem der Orbit komplett in der Sonne liegt

Die Geometrie hierzu zeigt Abbildung 3.4.

In dem rechtwinkligen Dreieck in dieser Abbildung lässt sich für den Betawinkel bei dem genannten Grenzfalle  $\beta_{\text{grenz}}$  folgende Gleichung ablesen:

$$\sin(\beta_{\text{grenz}}) = \frac{R_E}{a} = \frac{R_E}{R_E + h} \quad (3.5)$$

Ist  $|\beta| > \beta_{\text{grenz}}$  liegt der Orbit komplett in der Sonne, anderenfalls treten Eklipsen auf. Dabei fällt auf, dass  $\beta_{\text{grenz}}$  bei gegebener Bahnhöhe dem Öffnungswinkel  $\alpha_{\text{max}}$  für den Fall  $\beta = 0^\circ$  entspricht (siehe Gleichung (3.3)). Dies kann man dadurch erklären, dass beide Winkel im Prinzip dem Winkeldurchmesser der Erde aus der Höhe des

Raumfahrzeugs aus betrachtet entsprechen. Ist der Winkel  $\beta$  zwischen Orbitalebene und Sonneneinstrahlung größer als dieser, so befindet sich der komplette Orbit im Sonnenlicht und es können keine Eklipsen entstehen. Der Winkel  $\alpha_{max}$  entspricht dem Winkeldurchmesser des Erdschattens im Abstand des Bahnradius vom Erdmittelpunkt aus gesehen. Kehrt man die Blickrichtung um, erhält man wieder den Winkeldurchmesser der Erde vom Raumfahrzeug aus betrachtet. Alle drei Winkel sind also geometrisch betrachtet identisch.

Zur Berechnung von  $\alpha$  in Abhängigkeit von  $\beta$  findet sich in [3] folgende Gleichung:

$$\alpha = \arcsin \left( \frac{\sqrt{\left(\frac{R_E}{R_E+h}\right)^2 - \sin^2 \beta}}{\cos \beta} \right) \quad (3.6)$$

Diese Gleichung hat offensichtlich nur dann eine reelle Lösung, wenn der Radikant größer als null ist, also wenn

$$\begin{aligned} \left(\frac{R_E}{R_E+h}\right)^2 - \sin^2 \beta &> 0 \\ \Rightarrow |\beta| &< \arcsin \left(\frac{R_E}{R_E+h}\right) = \beta_{grenz} \end{aligned}$$

Logischerweise ergibt die Gleichung keine reelle Lösung, wenn der Betawinkel  $\beta_{grenz}$  überschreitet, also wenn keine Eklipse vorliegt. Für diesen Fall kann  $\alpha = 0^\circ$  gesetzt werden, so dass nach Gleichung (3.1)  $t_{ecl} = 0$  wird.

### 3.3 Einflüsse auf den Betawinkel und die Eklipsenzeiten

#### 3.3.1 Einleitende Betrachtungen

In der bisherigen Betrachtung wurde stets davon ausgegangen, dass der Betawinkel des betrachteten Orbits zu jedem Zeitpunkt bekannt ist. In diesem Abschnitt soll nun auf die Einflussfaktoren und die Berechnung von  $\beta$  eingegangen werden. Grundsätzlich hängt der Betawinkel von der Inklination  $i$  und der Rektaszension des aufsteigenden Knotens (RAAN)  $\Omega$  der betrachteten Umlaufbahn sowie von der relativen Position der Erde zur Sonne und damit von der Jahreszeit ab. Darüber hinaus spielen auch Bahnstörungen eine Rolle, die den Orbit über der Zeit verändern. Da Ergebnisse für komplette Missionen erzielt werden sollen, müssen diese in der endgültigen Analyse berücksichtigt werden. Für die folgende einführende Betrachtung werden sie jedoch zunächst vernachlässigt.

Geht man zunächst von einem kreisförmigen, äquatorialen Orbit aus, so ändert sich der Betawinkel im Laufe des Jahres. Dies liegt an der Neigung der Äquatorebene gegen die Ekliptik, die ca.  $23,5^\circ$  beträgt. Der Betawinkel beträgt zu den Tagundnachtgleichen, wenn die Sonne genau auf den Äquator scheint,  $0^\circ$  und zu den Sonnenwenden  $\pm 23,5^\circ$ .

Ein weiterer einfach zu betrachtender Fall sind die polaren Orbits. Hier hängt der Verlauf des Betawinkels über das Jahr jedoch zusätzlich von der RAAN des Orbits ab. Ist diese gleich  $0^\circ$  oder  $180^\circ$ , so wird der Betrag des Betawinkels zu den Sonnenwenden maximal. Zu diesen Zeitpunkten beträgt der Winkel zwischen der Sonneneinstrahlung und dem Äquator  $23,5^\circ$ , so dass der maximale Betawinkel  $90^\circ - 23,5^\circ = 66,5^\circ$  ist. Aus Gleichung (3.5) geht hervor, dass um die Sonnenwenden herum kurze Zeiten ohne Eklipsen auftreten, wenn die Bahnhöhe größer als 577 km ist. Ist die RAAN gleich  $90^\circ$  oder  $270^\circ$  liegt eine andere Situation vor. Der maximale Betawinkel tritt nun zu den Tagundnachtgleichen auf und erreicht im theoretischen optimalen Fall  $90^\circ$ . Es treten also auf allen Bahnhöhen Zeiten ohne Eklipsen auf, die deutlich länger sind, als bei einer RAAN von  $0^\circ$ . Die beiden hier beschriebenen Fälle stellen dabei Grenzfälle dar; liegt die RAAN zwischen diesen Werten, erreicht der Betawinkel entsprechend einen Maximalwert zwischen  $66,5^\circ$  und  $90^\circ$ .

### 3.3.2 Bahnstörungen

Um nun einen beliebigen Orbit betrachten zu können, müssen Bahnstörungen berücksichtigt werden. Die wichtigste dieser Störungen wird durch die Abweichung der Erde von der Kugelform hervorgerufen. Der deutlichste Unterschied zur Kugelform ist dabei die Abplattung der Erde aufgrund der Rotation; der äquatoriale Radius ist ca. 21 km größer als der polare. Diese rotationselliptische Gestalt bewirkt, dass der Erdgravitationsvektor nicht immer zum Mittelpunkt der Erde zeigt. Dadurch ändern sich die RAAN und das Argument des Perigäums (AoP) einer Bahn mit der Zeit. Diese Bahnstörung wird auch  $J_2$ -Störung genannt. Da hier nur kreisförmige Orbits betrachtet werden, kann die Änderung des AoP vernachlässigt werden. Für die Änderung der RAAN pro Tag für kreisförmige Orbits wird in [4] folgende Gleichung angegeben:

$$(\Delta\Omega)_{\text{Tag}} = -9,96^\circ \cdot \left(\frac{R_E}{a}\right)^{\frac{7}{2}} \cdot \cos i \quad (3.7)$$

Aus dieser Gleichung erkennt man, dass die  $J_2$ -Störung der RAAN für polare Bahnen mit  $i = 90^\circ$  verschwindet und für Bahnen geringer Inklination am größten wird, wobei sie bei  $i = 0^\circ$  keine Rolle spielt. Daher sind die oben ausgeführten Beispiele ohne Einschränkung gültig. Weiterhin geht aus der Gleichung hervor, dass die Störung der RAAN bei Inklinationen unter  $90^\circ$  negativ ist, so dass sich die Bahnebene des betrachteten Raumfahrzeugs entgegen der Umlaufrichtung der Erde um die Sonne dreht. Bei Inklinationen größer als  $90^\circ$  kann man durch die Bahnhöhe die Änderung der RAAN so einstellen, dass sie der Umlaufgeschwindigkeit um die Erde entspricht, so dass der Betawinkel nur geringen Schwankungen unterliegt. Für diesen Fall spricht man von einem Sonnensynchronen Orbit (SSO). Auf niedrigen Umlaufbahnen niedriger Inklination erreicht die Störung der RAAN Größenordnungen von mehreren Grad pro Tag.

Neben der  $J_2$ -Störung gibt es noch weitere Abweichungen der Erde von der Kugelgestalt, die jedoch deutlich weniger ausgeprägt sind und somit zu nur sehr langsamen Änderungen der Umlaufbahn führen. Da hier Missionsdauern von ungefähr einem Jahr betrachtet werden, können diese Störungen vernachlässigt werden. Darüber hin-

aus spielen insbesondere auf den sehr niedrigen Orbits Störungen durch die Restatmosphäre eine Rolle. Diese bewirkt bei jedem Raumfahrzeug einen aerodynamischen Widerstand, der zur Absenkung der Bahn und schließlich zum Wiedereintritt in die Erdatmosphäre führt. Für die folgenden allgemeinen Betrachtungen werden diese Störungen jedoch ebenfalls vernachlässigt.

### 3.3.3 Geometrie der Erdbahn

Betrachtet man nun zunächst einen beliebigen erdnahen Orbit mit einer beliebigen RAAN zu einem beliebigen Zeitpunkt, so wird klar, dass zur Berechnung des Beta-winkels die RAAN des betrachteten Orbits nicht ausreicht. Die RAAN ist als Winkel zwischen Frühlingspunkt und aufsteigendem Knoten definiert, was aber zunächst nichts über die Lage zur Sonne aussagt. Es muss also darüber hinaus der Winkel  $\theta$  zwischen Frühlingspunkt und Sonne in der Ekliptik bekannt sein. Die Bestimmung dieses Winkels für ein gegebenes Datum ist nicht ganz trivial, da der Erdborbit um die Sonne nicht kreisförmig sondern elliptisch ist, so dass sich die Erde mit unterschiedlichen Geschwindigkeiten bewegt. Darüber hinaus ist die Länge eines Kalenderjahres nicht konstant, weshalb sich die Erde zur gleichen Zeit in verschiedenen Jahren nicht an der gleichen Stelle befindet. Zusätzlich führt die Erde Präzessionsbewegungen um die Rotationsachse durch, weshalb die Position des Frühlingspunktes nicht konstant ist.

Um den Winkel zu einem gegebenem Zeitpunkt bestimmen zu können, müssen zunächst die Bahnparameter der Erde und deren zeitliche Änderungen zu einer bestimmten Epoche bekannt sein. Hierzu wird häufig die Epoche J2000.0 herangezogen, die für den 1. Januar 2000 um ca. 12 h Koordinierter Weltzeit (Coordinated Universal Time, UTC) definiert ist. Zu dieser Zeit ist das Julianische Datum, also die Anzahl der Tage nach dem 1. Januar 4713 v. Chr., gleich 2451545,0. In [5] werden die Bahnelemente der Erde (bzw. des Schwerpunkts des Erde-Mond-Systems) zur Epoche und die Änderungen der Elemente pro Jahr bis zum Jahr 2050 näherungsweise angegeben. Damit können die Bahnelemente zu einem beliebigen Zeitpunkt wie im Folgenden beschrieben näherungsweise bestimmt werden.

Zur Berechnung von  $\theta$  zu einem gegebenen Zeitpunkt muss zunächst die Anzahl der seit der Epoche J2000.0 vergangenen julianischen Jahrhunderte bestimmt werden, da die Änderungen der Bahnelemente in einem julianischen Jahrhundert gegeben sind. Algorithmen zur Umrechnung eines Gregorianischen Datums in ein Julianisches Datum finden sich an verschiedenen Stellen, zum Beispiel in [6], und werden hier nicht weiter behandelt. Zur Bestimmung des Winkels zwischen Sonne und Frühlingspunkt müssen die Exzentrizität  $e_E$ , die „mittlere Länge“ (Mean Longitude)  $ML_E$  und die Länge des Perihels (Longitude of Perihelion)  $\varpi_E$  der Erde zum gewünschten Zeitpunkt bestimmt werden. Der Index  $E$  bedeutet dabei, dass der Orbit der Erde gemeint ist; Bahnelemente von Orbits um die Erde werden ohne Index belassen. Die verschiedenen Größen hängen dabei im allgemeinen Fall wie folgt voneinander ab:

$$\begin{aligned}\varpi_E &= \Omega_E + \omega_E \\ ML_E &= M_E + \varpi_E = M_E + \Omega_E + \omega_E\end{aligned}$$

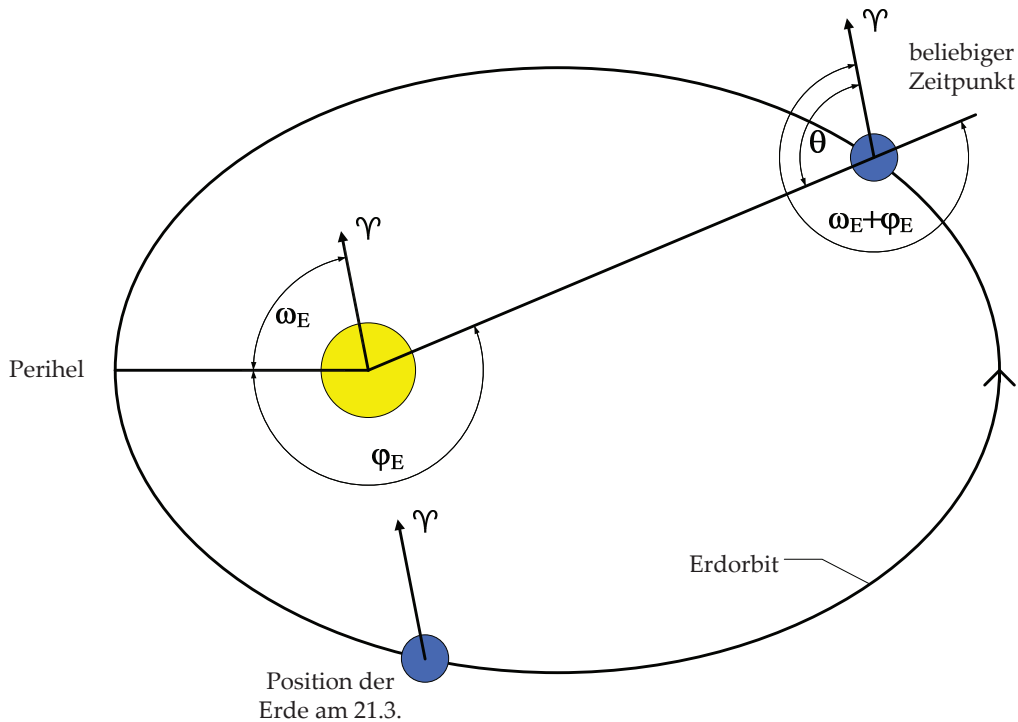
mit  $M_E$  als der mittleren Anomalie und  $\Omega_E$  als dem Argument des aufsteigenden Knotens (Longitude of the Ascending Node). Die Länge des Perihels  $\varpi_E$  entspricht dabei dem Argument des Perihels  $\omega_E$ , da als Referenzebene gerade die Ekliptik, also die Bahnebene der Erde, verwendet wird. Somit ist die Inklination gleich null und es gibt keine Knotenlinie, so dass das Argument des aufsteigenden Knotens  $\Omega_E = 0^\circ$  gesetzt werden kann. Somit kann die mittlere Anomalie berechnet werden zu

$$M_E = ML_E - \varpi_E = ML_E - \omega_E$$

Diese kann dann über ein einfaches numerisches Verfahren, zum Beispiel das NEWTON-Verfahren, in die wahre Anomalie  $\varphi_E$  umgerechnet werden. Ein Algorithmus dazu findet sich unter anderem in [7]. Der gesuchte Winkel zwischen Sonne und Frühlingspunkt ergibt sich dann zu

$$\theta = \begin{cases} \varphi_E + \omega_E + 180^\circ, & \varphi_E + \omega_E < 180^\circ \\ \varphi_E + \omega_E - 180^\circ, & \varphi_E + \omega_E \geq 180^\circ \end{cases} \quad (3.8)$$

wie aus Abbildung 3.5 ersichtlich ist.



**Abbildung 3.5:** Geometrie des Erdborbits zur Bestimmung von  $\theta$  mit der Position der Erde am 21.3. zur Definition des Frühlingspunktes, die Exzentrizität des Erdborbits ist stark übertrieben (nicht maßstabsgetreu)

Sind der Winkel  $\theta$  und die RAAN eines Orbits um die Erde bekannt, so lässt sich die Orientierung der Bahnebene zu einem gegebenem Zeitpunkt im Bezug zur Sonne bestimmen. Ist darüber hinaus die Inklination des Orbits bekannt, kann der Betawinkel unter Berücksichtigung der Neigung der Äquatorebene gegen die Ekliptik  $i_{Äq,Ekl} =$

23,5° für den gegebenen Zeitpunkt berechnet werden. In [3] wird dazu die folgende aus Überlegungen der sphärischen Geometrie hergeleitete Gleichung angegeben:

$$\sin \beta = \cos \theta \sin \Omega \sin i - \sin \theta \cos i_{\ddot{A}q,Ekl} \cos \Omega \sin i + \sin \theta \sin i_{\ddot{A}q,Ekl} \cos i \quad (3.9)$$

### 3.3.4 Verlauf des Betawinkels und der Eklipsenzeiten für beliebige Orbits

#### Orbits mit Inklinationen zwischen 0° und 90°

Aus den vorherigen Überlegungen lassen sich der Betawinkel und somit die Eklipsendauer für jeden beliebigen LEO zu jedem beliebigen Zeitpunkt bis 2050 näherungsweise bestimmen. Betrachtet man nun den Verlauf des Betawinkels bzw. der Eklipsenzeiten über einen längeren Zeitraum, muss das Zusammenspiel von der jahreszeitlich bedingten Position der Sonne, der Einstrahlrichtung der Sonne im Bezug auf den Äquator und die Änderung der RAAN des Orbits aufgrund der  $J_2$ -Störung in Betracht gezogen werden. Die für CLAVIS angedachte Missionsdauer ist ca. ein Jahr, was für die folgenden Betrachtungen als Zeitrahmen verwendet wird. Da die  $J_2$ -Störung eine Drehung einer Bahnebene mit  $i < 90^\circ$  entgegen der Drehrichtung der Erde um die Sonne von mehreren Grad pro Tag hervorruft, gibt es über ein Jahr eine Vielzahl von Maxima und Minima des Betawinkels und damit der Eklipsendauer.

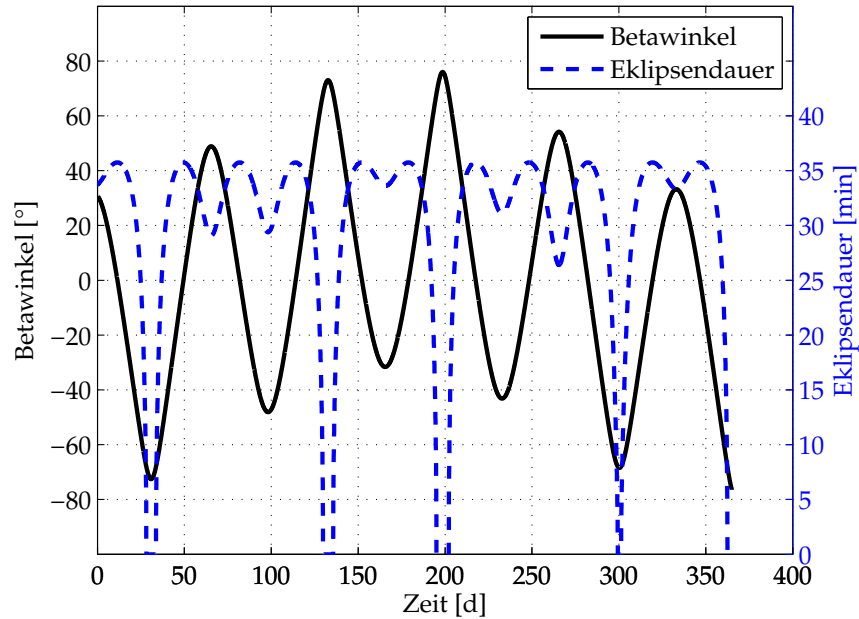
Nimmt man zum Beispiel eine Inklination von 50° und eine Bahnhöhe von 500 km an, so ergibt sich nach Gleichung (3.7) eine Änderung der RAAN von  $\Delta\Omega = -4,92^\circ$  pro Tag aufgrund der  $J_2$ -Störung. Durch die Drehung der Erde um die Sonne ändert sich die Einstrahlrichtung der Sonne auf einen beliebigen Orbit dabei um  $\Delta\varphi = \frac{360^\circ}{365\text{d}} = +0,99^\circ$  pro Tag, wenn man vereinfachend einen kreisförmigen Erdorbit annimmt. Mit diesen Werten lässt sich die Dauer zwischen einem Maximum und einem Minimum des Betawinkels näherungsweise berechnen. Dazu geht man davon aus, dass beide Einflüsse zusammen genommen eine Winkeldistanz von  $180^\circ$  zwischen einem Maximum und einem Minimum des Betawinkels aufbauen müssen. Für den hier betrachteten Fall mit  $i < 90^\circ$  wirken beide Einflüsse in unterschiedliche Richtungen, so dass die Abfolge von lokalen Extremwerten des Betawinkelverlaufs beschleunigt wird. Daher müssen sie mit unterschiedlichem Vorzeichen berücksichtigt werden. Für die ungefähre Dauer zwischen zwei Extremstellen des Betawinkels  $t_{min,max}$  ergibt sich

$$t_{min,max} \approx \frac{180^\circ}{\Delta\varphi - \Delta\Omega} = \frac{180^\circ}{0,99^\circ/\text{d} + 4,92^\circ/\text{d}} = 30,5\text{ d}$$

Nach dieser stark vereinfachten Abschätzung ergeben sich also ca. 12 Extremstellen des Betawinkels in einem Jahr. Da die Eklipsendauer bei  $\beta = 0^\circ$  maximal wird und an jeder Extremstelle des Betawinkels minimal ist, besitzt sie ungefähr doppelt so viele Extremstellen pro Jahr. Für niedrigere Inklinationen und niedrigere Bahnen erhöht sich die Anzahl der Zyklen, wobei im erdnahen Bereich die Inklination den wichtigeren Einfluss darstellt.

Die folgende Abbildung 3.6 zeigt die Verläufe des Betawinkels und der Eklipsendauer für den oben genannten Orbit mit einer RAAN von  $0^\circ$  zum Startzeitpunkt über

ein Jahr, wobei als Startzeitpunkt der 01.01.2011 willkürlich gewählt wurde. Die Abbildung wurde mit Hilfe von MATLAB erstellt, wobei der oben beschriebene Berechnungsablauf verwendet wurde, bei der die  $J_2$ -Störung als einziger Störeinfluss berücksichtigt wird. Die verwendeten Quellcodes finden sich im Anhang A.

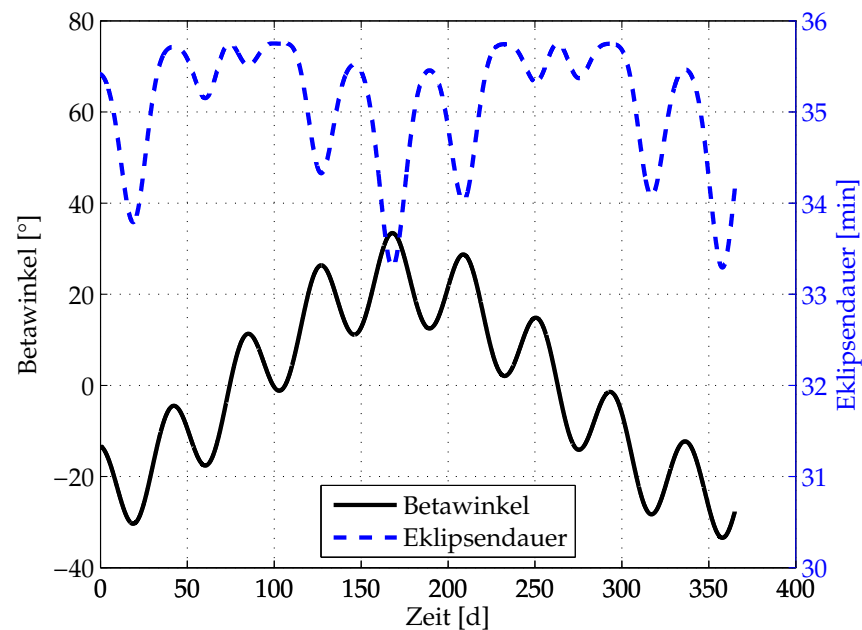


**Abbildung 3.6:** Verläufe von Betawinkel und Eklipsendauer für einen Orbit mit  $i = 50^\circ$  und  $h = 500$  km über ein Jahr mit dem Startzeitpunkt 01.01.2011 0 h unter Berücksichtigung der  $J_2$ -Störung

Anhand dieser Abbildung erkennt man die oben beschriebenen Zusammenhänge zwischen den Extremwerten von Betawinkel und Eklipsendauer. Der Verlauf des Betawinkels ähnelt einer Überlagerung zweier Schwingungen. Dabei kann die Änderung der Einstrahlrichtung des Sonnenlichts zum Äquator im Laufe des Jahres als die „niederfrequente“ und die durch die  $J_2$ -Störung und die Drehung der Erde um die Sonne hervorgerufene Änderung des Einstrahlwinkels zur Bahnebene als die „höherfrequente“ angesehen werden. Man erkennt, dass der Betawinkel im Sommer eher positiv und im Winter eher negativ ist, was für Inklinationen unter  $90^\circ$  generell gilt. Der genaue Wert der Minima der Eklipsendauer ergibt sich aus der Überlagerung der beiden Einflüsse. Ist während eines Zeitraumes  $|\beta| > \beta_{\text{grenz}}$  (siehe Abschnitt 3.2), gibt es während dieser Zeit keine Eklipsen. Darüber hinaus ist ersichtlich, dass die maximale Eklipsendauer praktisch konstant ist.

Abbildung 3.7 zeigt ein ähnliches Diagramm wie Abbildung 3.6 für eine Inklination von  $10^\circ$ . Es zeigt sich, dass die Änderungen des Betawinkels und damit der Eklipsendauer deutlich geringer ausfallen. Der maximal mögliche Betawinkel ergibt sich hierbei zu  $\beta_{\text{max}} = 23,5^\circ + i = 33,5^\circ$ . Daher gibt es bei einer solchen Inklination auch keine Zeiten ohne Eklipse. Wendet man wieder den Vergleich zu Schwingungen an, zeigt sich dass die kurzfristigen Änderungen des Betawinkels eine höhere Frequenz und ei-





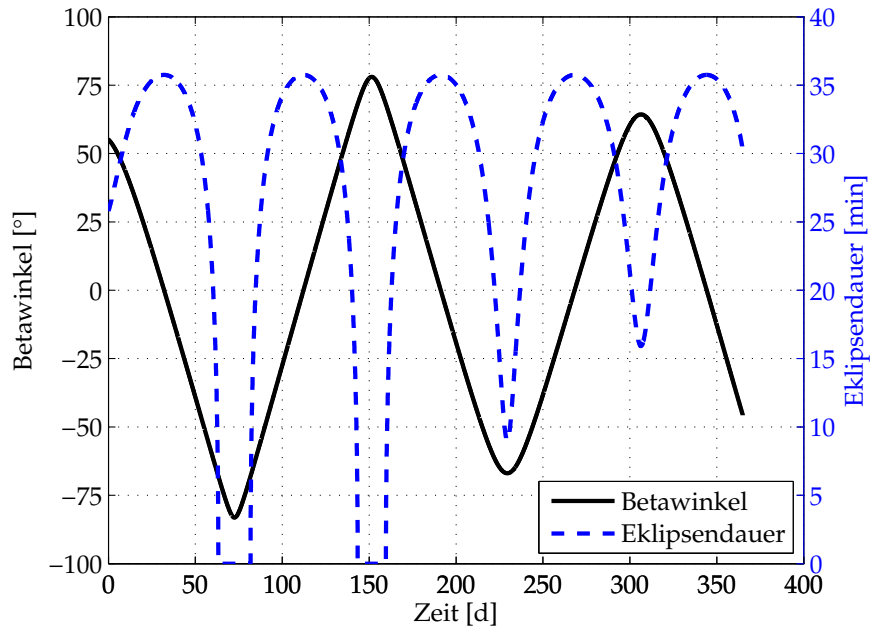
**Abbildung 3.7:** Verläufe von Betawinkel und Eklipsendauer für einen Orbit mit  $i = 10^\circ$  und  $h = 500$  km über ein Jahr mit dem Startzeitpunkt 01.01.2011 0 h unter Berücksichtigung der  $J_2$ -Störung

ne niedrigere Amplitude als bei der höheren Inklination besitzen. Außerdem fällt auf, dass die maximale Eklipsendauer nicht mehr konstant ist, sondern im Sommer und Winter niedriger als der theoretische Maximalwert ist. Dies liegt daran, dass die Einstrahlungsrichtung der Sonne für begrenzte Zeiträume niemals in der Orbitalebene liegt ( $\beta$  wird niemals null), da die Neigung der Äquatorbene gegen die Ekliptik größer als die Inklination ist.

Abbildung 3.7 zeigt die beiden Verläufe für einen Orbit mit einer Inklination von  $80^\circ$ . Hierbei sieht man, dass die Änderungen des Betawinkels aufgrund der  $J_2$ -Störung im Vergleich zu den vorherigen Fällen deutlich langsamer vonstatten gehen, aber einen größeren Bereich abdecken. Dadurch ist die jährliche Änderung durch die Neigung des Äquators gegen die Ekliptik kaum noch zu erkennen. Im Fall eines betragsmäßig besonders großen Betawinkels ergeben sich verhältnismäßig lange Zeiten ohne Eklipsen.

### Sonnensynchrone Orbits

Sonnensynchrone Orbits stellen einen wichtigen Sonderfall dar, da die Lage der Orbitalebene zur Sonne näherungsweise konstant bleibt. So lassen sich Orbits finden, auf denen Raumfahrzeuge niemals in den Erschatten gelangen und stets ungefähr die gleiche Tageszeit in den überflogenen Gebieten herrscht, so dass Aufnahmen von Erdbeobachtungssatelliten einfach verglichen werden können. Wie bereits erwähnt besitzen SSOs eine Inklination von mehr als  $90^\circ$ . Mit Hilfe der mittleren Winkelgeschwindigkeit der Erde um die Sonne von ungefähr  $0,99 \frac{^\circ}{\text{d}}$  und Gleichung (3.7) kann man für



**Abbildung 3.8:** Verläufe von Betawinkel und Eklipsendauer für einen Orbit mit  $i = 80^\circ$  und  $h = 500$  km über ein Jahr mit dem Startzeitpunkt 01.01.2011 0h unter Berücksichtigung der  $J_2$ -Störung

kreisförmige Orbits die Inklination in Abhängigkeit der großen Halbachse angeben zu

$$i = \arccos \left( -0,09896 \cdot \left( \frac{a}{R_E} \right)^{\frac{7}{2}} \right) \quad (3.10)$$

Für den hier relevanten Höhenbereich zwischen 200 km und 650 km ergeben sich dadurch Inklinationen zwischen  $96^\circ$  und  $98^\circ$ .

Zur Beschreibung von SSOs wird die Angabe der RAAN zu einem bestimmten Zeitpunkt häufig durch die anschaulichere „Local Time of Ascending Node“ (LTAN) bzw. die „Local Time of Descending Node“ (LTDN) angegeben. Dabei handelt es sich um die wahre Ortszeit des Punktes unter dem aufsteigenden (LTAN) bzw. absteigenden Knoten (LTDN) der Bahn. In der weiteren Betrachtung wird hier die LTDN verwendet. Dieser Wert ist aufgrund der Drehung der Orbitalebene näherungsweise konstant über die Zeit, wobei aufgrund der Elliptizität des Erdborbits geringe Abweichungen von einigen Minuten auftreten. Ist die LTDN 0h oder 12h ist  $\beta \approx 0^\circ$ , so dass die Eklipsendauer näherungsweise konstant ist. Ist die LTDN hingegen 6h oder 18h, fliegt der Satellit gerade über die Tag-Nachtgrenze, so dass er über lange Zeiten in der Sonne ist. Je nach Bahnhöhe gibt es aber Orbits, auf denen auch Eklipsen auftreten.

Die LTDN kann aus dem aktuellen Datum und der aktuellen RAAN berechnet werden. Dazu wird ein ähnliches Verfahren wie bei der Bestimmung des Winkels  $\theta$  angewendet (siehe Abschnitt 3.3.3). Zur einfacheren Berechnung der LTDN kann sie als Winkel angegeben werden, wobei 0h  $0^\circ$  und 24h  $360^\circ$  entsprechen. Die LTDN ergibt sich mit Hilfe der lokalen Zeit des Frühlingspunktes  $LT_\gamma$  und der RAAN  $\Omega$  des betrachteten

Orbits um die Erde zum betrachteten Zeitpunkt zu

$$LTDN = LT_{\gamma} + \Omega + 180^{\circ}$$

Die lokale Zeit des Frühlingspunktes berechnet sich unter Berücksichtigung des Winkels zwischen Frühlingspunkt und Erde  $u_E = \varphi_E + \omega_E$  zu

$$LT_{\gamma} = 360^{\circ} - u_E = 360^{\circ} - \varphi_E - \omega_E$$

Daraus folgt für die LTDN

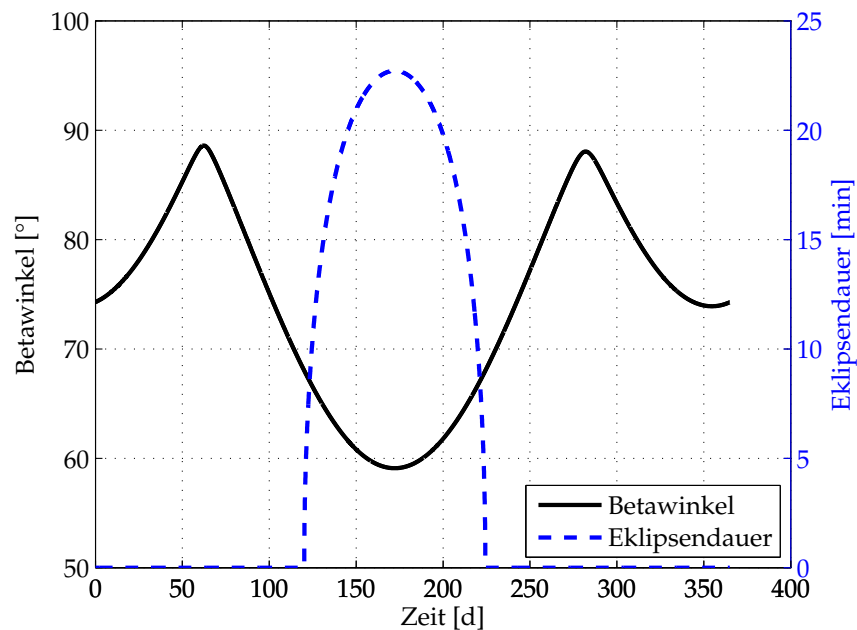
$$LTDN = 180^{\circ} - \varphi_E - \omega_E + \Omega \quad (3.11)$$

Die Bestimmung der entsprechenden Werte der Erdbahn ist in Abschnitt 3.3.3 beschrieben.

Betrachtet man nun zunächst einen SSO mit einer LTDN von 0 h oder 12 h, so ist wie schon erwähnt der Betawinkel stets ungefähr gleich  $0^{\circ}$  und somit die Eklipsendauer immer in der Nähe des theoretischen Maximums. Dies gilt aber nur näherungsweise, da aufgrund der Elliptizität des Erdborbits die Geschwindigkeit der Erde um die Sonne im Gegensatz zur durch die  $J_2$ -Störung hervorgerufenen Drehung des Orbits nicht konstant ist. Es ergeben sich also Abweichungen des Betawinkels von wenigen Grad und damit Abweichungen der Eklipsendauer von wenigen Sekunden im Laufe des Jahres.

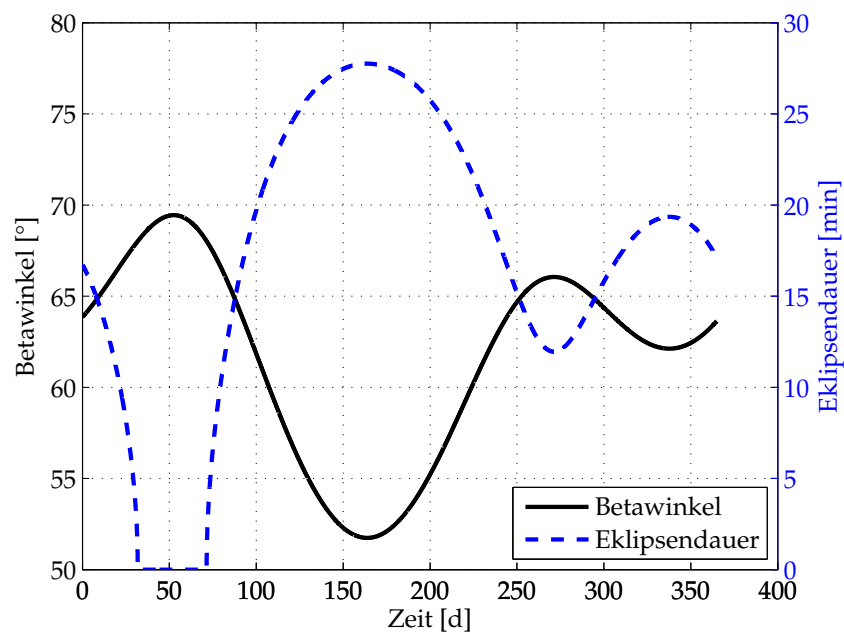
Im Gegensatz hierzu steht der Fall mit einer LTDN von 6 h oder 18 h. Hierbei ist der Betawinkel stets relativ groß, kann allerdings nur zu zwei Zeitpunkten im Jahr, wenn der Winkel zwischen Sonneneinstrahlung und Äquatorebene gerade der Differenz zwischen der Inklination des SSO und  $90^{\circ}$  entspricht, das theoretische Maximum von  $90^{\circ}$  erreichen. Es ergeben sich also lange Zeiten ohne Eklipse. Da der Betawinkel als der kleinste Winkel zwischen Sonneneinstrahlungsrichtung und Orbitalebene definiert ist, ergibt sich hier ein wichtiger Unterschied zu dem ersten Sonderfall. Ist die LTDN ungefähr gleich 6 h oder 18 h befindet sich dieser kleinste Winkel im Sommer bzw. im Winter zu den Sonnenwenden zwischen der Sonneneinstrahlung und dem Pol der Bahn. Im Gegensatz dazu ist der kleinste Winkel bei einer LTDN von ungefähr 0 h oder 12 h zwischen der Sonneneinstrahlung und dem nähergelegenen Knoten. Daraus ergibt sich der Unterschied, dass der Betawinkel im Fall einer LTDN von 6 h oder 18 h stärker schwankt, da entweder im Sommer ( $LTDN = 6$  h) oder im Winter ( $LTDN = 18$  h) von der Einstrahlrichtung der Sonne aus betrachtet der nächste Punkt des Orbits dieserseits des nähergelegenen Pols der Erde liegt, da ein SSO immer eine Inklination von etwas mehr als  $90^{\circ}$  besitzt. Somit wird der Betawinkel zu diesen Zeitpunkten relativ klein, so dass hier je nach Bahnhöhe kürzere oder längere Eklipsen auftreten können.

Abbildung 3.9 zeigt beispielhaft den Verlauf von Betawinkel und Eklipsendauer über ein Jahr für einen SSO in 500 km Höhe. Als Startzeitpunkt wird hier wieder der 01.01.2011 verwendet, wobei die RAAN auf ca.  $10^{\circ}$  gesetzt wurde, so dass sich eine LTDN von 6 h ergibt. Man erkennt einen Zeitraum im Sommer, während dem Eklipsen auftreten, mit einer maximalen Eklipsendauer von ca. 22 Minuten bei besonders niedrigem Betawinkel. Der Betawinkel erreicht seine Maxima grob gesagt in der Nähe



**Abbildung 3.9:** Verläufe von Betawinkel und Eklipsendauer für einen SSO mit  $h = 500$  km und  $LTDN = 6$  h über ein Jahr mit dem Startzeitpunkt 01.01.2011 0 h unter Berücksichtigung der  $J_2$ -Störung

der Tagundnachtgleichen, da für diesen relativ niedrigen SSO die Inklination nicht so stark von  $90^\circ$  abweicht. Orbits, in denen keine Eklipse auftritt, müssen eine Bahnhöhe



**Abbildung 3.10:** Verläufe von Betawinkel und Eklipsendauer für einen SSO mit  $h = 500$  km und  $LTDN = 7:30$  h über ein Jahr mit dem Startzeitpunkt 01.01.2011 0 h unter Berücksichtigung der  $J_2$ -Störung

von über 1400 km besitzen.

Abschließend zeigt Abbildung 3.10 den Verlauf von Betawinkel und Eklipsendauer für eine LTDN von 7:30 h mit dem gleichen Startzeitpunkt. Man erkennt einen kurzen Zeitraum ohne Eklipsen bei hohem Betawinkel kurz vor der Frühlings-Tagundnachtgleiche. Außerdem zeigt sich, dass aufgrund der mittleren LTDN die Symmetrie der Verläufe verloren geht. Steigert man die LTDN weiter, so behalten die Verläufe im Prinzip ihre Form bei, jedoch verringern sich die Unterschiede zwischen Minimal- und Maximalwerten, bis sie bei einer LTDN von 12 h praktisch verschwinden.

# Kapitel 4

## Energieversorgungssystem

### 4.1 Gesamtsystem

Wie in Kapitel 2 bereits erwähnt, besitzt CLAVIS zur Erzeugung elektrischer Energie sowohl in der Standardkonfiguration als auch in der AISat-Konfiguration fünf Solarpaneele an allen Seiten bis auf der erdzugewandten Seite. Die Solarpaneele werden von der Firma SPECTROLAB hergestellt und bestehen in der Standardkonfiguration jeweils aus zwei Arrays, die wiederum aus sieben in Reihe geschalteten Solarzellen bestehen. Abbildung 4.1 zeigt ein solches Solarpaneel in der Draufsicht.

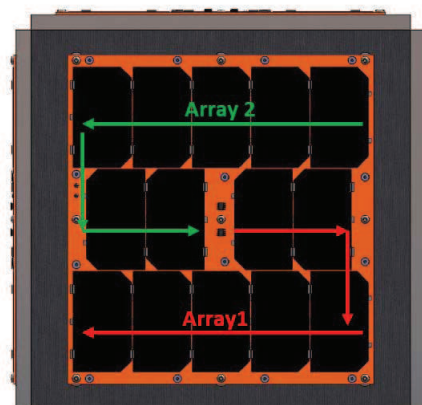


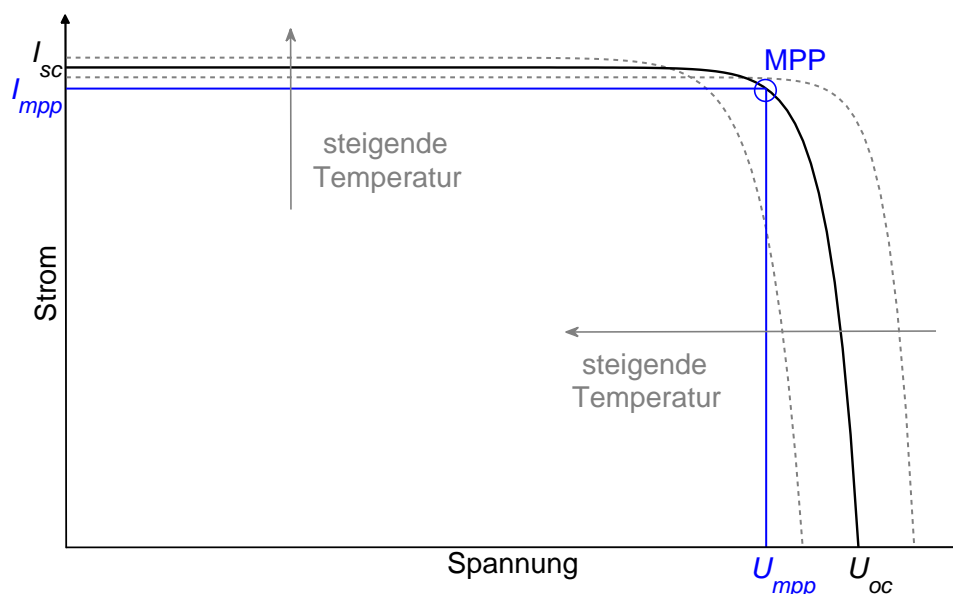
Abbildung 4.1: Solarpaneele von CLAVIS aus [8]

Man erkennt die einzelnen Solarzellen als schwarze Rechtecke mit gefasteten Ecken. Ihre Fläche beträgt jeweils  $26,62\text{ cm}^2$ . Wie bereits erwähnt, werden für die AISat-Mission etwas kleinere Paneele mit jeweils zwei Arrays aus jeweils sechs Solarzellen, also insgesamt zwölf Solarzellen pro Paneel, verwendet. Für zukünftige CLAVIS-Missionen wird auch der Einsatz von Solarpaneelen mit mehr als 14 Zellen geprüft (siehe auch Abschnitt 10.3.2). Die folgenden Darstellungen beziehen sich auf das Energieversorgungssystem der AISat-Konfiguration.

Bei den Solarzellen handelt es sich um sogenannte „Ultra Triple Junction“ Solarzellen, wobei eine einzige Zelle im Grunde aus drei in Serie geschalteten, übereinanderliegen-

den Zellen besteht. Der Sinn dieses Aufbaus liegt darin, dass Photonen verschiedener Wellenlängen von verschiedenen Materialien zur Erzeugung von elektrischer Energie genutzt werden können. Die obersten Schichten sind daher durchlässig für bestimmte Wellenlängen, so dass die darunterliegenden Zellen ebenfalls zur Energieerzeugung beitragen. Eine solche auf Deutsch auch „Tandem-Solarzelle“ genannte Zelle kann somit einen höheren Wirkungsgrad erzielen als eine herkömmliche. Im vorliegenden Fall besteht die oberste Zelle aus  $\text{GaInP}_2$ , die mittlere aus GaAs und die unterste aus Ge. Zur Verarbeitung der erzeugten Energie wird ein sogenanntes „Deployed Electronic Power System“ (DEPS) verwendet, das von der Firma CLYDESPACE hergestellt wird. Das Gesamtsystem wird auf Grundlage der entsprechenden Dokumentation [8] im Folgenden beschrieben.

Die während einer Sonnenlichtperiode in einem Solarpaneel generierte elektrische Energie gelangt zunächst in einen von sechs unabhängigen sogenannten „Battery Charge Regulators“ (BCR), in denen Spannung und Strom zum Aufladen der Batterie und zur Versorgung der Verbraucher geregelt werden. Dabei sind jeweils zwei Strings von zwei gegenüberliegenden Paneelen mit einem BCR verbunden, der 8 W aufnehmen kann. Da niemals Paneele auf gegenüberliegenden Seiten gleichzeitig angestrahlt werden, wird die Leistung von 8 W praktisch nie überschritten. Darüber hinaus findet in den BCRs das sogenannte „Maximum Power Point Tracking“ (MPPT) statt. Zur Erläuterung dieses Verfahrens zeigt Abbildung 4.2 eine typische Strom-Spannungs-Kennlinie einer Solarzelle. Man erkennt die eigentliche Kennlinie als



**Abbildung 4.2:** Strom-Spannungs-Kennlinie einer Solarzelle

schwarze durchgezogene Linie mit dem Kurzschlussstrom  $I_{sc}$  als dem maximalen Strom und der Leerlaufspannung  $U_{oc}$  als der maximalen Spannung. Aufgrund des Verlaufes der Kurve ergibt sich die maximal mögliche Leistung an einem Punkt in der Nähe des „Knies“ der Kurve, der hier als „Maximum Power Point“ (MPP) bezeichnet ist, wobei die dazugehörige Spannung als  $U_{mpp}$  und der dazugehörige Strom als  $I_{mpp}$

bezeichnet werden. Ebenfalls aus der Abbildung ersichtlich ist, dass die Kennlinie bei sich ändernden Temperaturen ihre Gestalt ändert, was durch die gestrichelten Linien dargestellt wird. Daher muss der MPP regelmäßig neu bestimmt werden. Dazu wird aus der vom Solarpaneel bezogenen Energie der MPP ungefähr alle 2,5 s aufgenommen. Da während der Berechnung des MPP die Leerlaufspannung angelegt wird, ergeben sich hierbei jedoch Verluste.

Die durch den BCR eingestellte elektrische Energie der Solarpaneele wird nun zum Versorgen der Satellitensysteme und der Nutzlast, sowie zum Aufladen der Batterie verwendet. Dazu stehen die „Power Conditioning Modules“ (PCM) und die „Power Distribution Modules“ (PDM) zur Verfügung. Diese können die Verbraucher unreguliert von der Batterie oder mit Bussystemen mit Nennspannungen von 3,3 V, 5 V oder 12 V mit elektrischer Energie versorgen. Der unregulierte Bus wird im Energieversorgungssystem von CLAVIS nicht verwendet. Aufgrund der Einstellung der Spannung verursachen die anderen Bussysteme gewisse Verluste, die bei den beiden Systemen mit niedrigerer Versorgungsspannung geringer sind, weshalb diese bevorzugt verwendet werden. Verschiedene Sicherheitssysteme sind vorhanden, um die Energieversorgung bei zu hohen Stromflüssen abzuschalten.

Die Batterieeinheit besteht aus zwei Lithium-Polymer-Batterien, die jeweils aus zwei einzelnen galvanischen Zellen bestehen. Ihre maximale Gesamtkapazität beträgt ca. 4,64 Ah (siehe Abschnitt 4.4). Die Batterie wird während der Sonnenlichtzeiten aufgeladen, bis eine Maximalspannung von 8,26 V sowie die maximale Kapazität erreicht sind und während der Schattenzeiten entladen, wobei eine Minimalspannung von 6,2 V nicht unterschritten werden sollte [9]. Damit ihre Betriebstemperatur nicht unter 0 °C sinkt, besitzt die Batterie einen Heizer und eine entsprechende Temperatursensorik. Während des Ladevorgangs wird die Batterie mit einem konstanten Strom versorgt, wobei die Versorgung im MPP der Solarpaneele stattfindet, bis die Maximalspannung, auch „End-of-Charge-Voltage“ genannt, erreicht ist. Im allgemeinen Fall ist die Kapazität dann noch nicht voll wiederhergestellt, so dass mit abnehmendem Strom bei konstanter Spannung weitergeladen wird („Taper Charge“), bis die maximale Kapazität erreicht ist. Um sicherzugehen, dass die Batterie vollständig geladen wird, wird ihr dabei etwa 10 % mehr Energie zugeführt, als ihr entnommen wurde. Dieser Wert wird auch als „Battery Return Factor“ bezeichnet, der dann mit  $RF = 1,1$  angegeben wird. Während dieses letzten Teils des Aufladevorgangs wird die überschüssige Energie in den Solarpaneelen als Wärme dissipiert.

Beim Entladen der Batterie sollte die minimale Spannung niemals unterschritten werden, was durch ein spezielles Sicherheitssystem sichergestellt wird. Entlädt man die Batterie jedoch öfters komplett, so verringert sich die maximale Kapazität mit jedem Ladezyklus schneller, als wenn sie nur bis zu einem gewissen Maße entladen wird. Man definiert daher die sogenannte „Depth of Discharge“ (DoD) als den Anteil der Gesamtkapazität, der beim Entladen entnommen wird. Damit sich die Kapazität der Batterie während der auf LEO bis zu 6000 Ladezyklen im Jahr nicht übermäßig verringert, wird die DoD auf 20 % begrenzt. Das gesamte DEPS benötigt zu jedem Zeitpunkt eine elektrische Leistung von ca. 0,1 W.



## 4.2 Energiebudget

Damit abgeschätzt werden kann, ob die erzeugte Energie ausreicht, um alle Systeme ausreichend zu versorgen, wird im Folgenden ein vereinfachtes Energiebudget für ein Worst-Case-Szenario an Hand des vorhandenen Energiebudgets von CLAVIS aufgestellt [10]. Da die einzelnen Systeme für die Standardkonfiguration des CLAVIS noch nicht genau definiert sind, wird hier zunächst das Energiebudget für die AISat-Konfiguration betrachtet.

Grundsätzlich sind verschiedene Betriebsmodi denkbar. Der Großteil der Zeit wird bei normaler Funktion des Satelliten dem Betrieb der Nutzlast, hier also der AIS-Empfänger, gewidmet. Ein geringerer Anteil der verfügbaren Zeit wird verwendet, um mit den Bodenstationen zu kommunizieren und gewonnene Daten zu übertragen. Die Zeit für diesen Modus ist dabei durch die Bodenkontaktzeiten begrenzt, die auf LEO stets nur einen kleinen Teil eines Orbits ausmachen, wenn man von einer einzigen Bodenstation ausgeht. Ein weiterer denkbarer Modus wäre eine Art Standby-Betrieb, bei dem weder die Nutzlast noch das Kommunikationssystem betrieben werden, was aber im Vergleich zu den anderen Modi unkritisch in Hinsicht auf die zur Verfügung stehende Energie ist. Darüber hinaus gibt es eine Phase zu Beginn der Mission, in der eine relativ große Stromstärke benötigt wird, um den Entfaltungsmechanismus der Helixantenne auszulösen. Da diese Phase jedoch einmalig und relativ kurz ist, kann eine stärkere Belastung der Batterie in Kauf genommen werden, so dass diese Phase hier nicht weiter betrachtet wird.

Die folgende Tabelle 4.1 zeigt die einzelnen Systeme des Satelliten, unterteilt in Subsysteme von CLAVIS und die Nutzlast, mit ihrer jeweiligen maximalen Leistungsaufnahme sowie den durchschnittlichen Auslastungen bei den beiden verschiedenen Betriebsmodi und den entsprechenden durchschnittlichen Leistungsaufnahmen.

Grundsätzlich erkennt man anhand der Tabelle, dass der Großteil der CLAVIS-Systeme, bis auf den UHF-Transmitter, in beiden betrachteten Modi mit der gleichen Auslastung betrieben wird. Dabei laufen die meisten Komponenten die ganze Zeit unter Volllast, bis auf die Magnet Torquer und den Heizer der Batterie. Die Torquer sind nur die Hälfte der Zeit in Betrieb, da während der verbleibenden Zeit das Magnetfeld gemessen wird. Während dieser Zeit müssen die Torquer ausgeschaltet werden, um die Messung nicht zu verfälschen. Der Heizer der Batterie wird automatisch dann angestellt, wenn die Batterietemperatur nahe  $0^{\circ}\text{C}$  kommt, was lediglich gegen Ende einer Schattenperiode auftreten kann. Die hier veranschlagten 10 % Auslastung stellen also eine konservative Annahme dar. Der Ruhestrom der Batterie und der Verbrauch des Heizers sind hier für beide Batterien zusammen aufgeführt.

Im Kommunikationsmodus läuft zusätzlich zu den anderen CLAVIS-Systemen außerdem der UHF-Transmitter zur Übertragung von Daten zu einer Bodenstation auf voller Leistung, während die Nutzlast mit Ausnahme des UHF-Beacons ausgeschaltet ist. Dieses Beacon (Funkfeuer) ist ein Standardsignal, das durchgehend vom Satelliten zur Erkennung gesendet wird, wobei 10 % der Leistung des Senders ausreichen. Der UHF-Empfänger läuft ebenfalls grundsätzlich immer, damit der Satellit im Notfall

<b>Komponente</b>	<i>max. Leistung</i> [W]	<b>Nutzlastbetrieb</b>		<b>Kommunikation</b>	
		<i>Auslastung</i> [%]	<i>Leistung</i> [W]	<i>Auslastung</i> [%]	<i>Leistung</i> [W]
<i>CLAVIS</i>					
Computer	0,330	100	0,330	100	0,330
DEPS	0,099	100	0,099	100	0,099
Magnet Torquer	0,690	50	0,345	50	0,345
UHF-Empfänger	0,231	100	0,231	100	0,231
UHF-Transmitter	4,950	0	0	100	4,950
TM-Switchboard	0,150	100	0,150	100	0,150
Batterie Ruhestrom	0,200	100	0,200	100	0,200
Batterieheizer	1,000	10	0,1	10	0,1
Batterie PCB	0,099	100	0,099	100	0,099
IMU	0,368	100	0,368	100	0,368
<i>Nutzlast</i>					
AAU-Empfänger I	0,462	100	0,462	0	0
AAU-Empfänger II	1,107	20	0,221	0	0
DLR-Empfänger	6,000	20	1,200	0	0
UHF-Beacon	1,680	10	0,168	10	0,168
Kamera	0,251	5	0,013	0	0
<b>Summe</b>	<b>17,617</b>		<b>3,985</b>		<b>7,040</b>

**Tabelle 4.1:** Betriebsmodi der AISat-Konfiguration mit Leistungsaufnahme und Auslastungen der Subsysteme

stets erreicht werden kann.

Während des Betriebes der Nutzlast wird der erste Empfänger der Aalborg Universität (AAU) dauerhaft betrieben. Die anderen Empfänger können zusätzlich angeschaltet werden, allerdings jeweils nur einer und nur für maximal 20 Minuten pro Tag. In dieser Zeit erzeugen diese Empfänger eine so große Datenmenge, dass die Bodenkontaktzeiten eines Tages benötigt werden, um die Daten an die Bodenstation zu senden. Hier wird davon ausgegangen, dass ein Orbit ca. 100 Minuten dauert und dass beide AIS-Empfänger während eines Orbits im schlechtesten Fall jeweils 20 Minuten, also 20 % der Zeit, betrieben werden. Zusätzlich ist eine Kamera an Bord, mit der einzelne Aufnahmen gemacht werden sollen, weshalb diese eine sehr geringe Auslastung besitzt.

Für die weitere Betrachtung wird ein Worst-Case-Szenario angenommen, in dem der Satellit während eines Orbits ungefähr 90 % der Zeit im Nutzlastbetriebsmodus ist und ungefähr 10 % der Zeit im Kommunikationsmodus. Dieser Wert stimmt mit der maximalen Bodenkontaktdauer während eines Orbits von ungefähr neun Minuten (siehe Abschnitt 9.3) grob überein. Mit diesen Faktoren und den in Tabelle 4.1 gegebenen Werten ergibt sich für einen Orbit mit maximaler aufgewendeter Energie des Satellitensystems eine durchschnittlich aufgenommene Leistung von

$$P_{auf} = 0,9 \cdot 3,985 \text{ W} + 0,1 \cdot 7,040 \text{ W} \approx 4,29 \text{ W} \quad (4.1)$$

Dieser Wert wird in den folgenden Betrachtungen stets verwendet. Dabei muss natürlich beachtet werden, dass die Leistungsaufnahme aufgrund der unterschiedlichen Instrumente nicht konstant über die Zeit ist.

### 4.3 Solarpaneele

Zur Berechnung der durch die Solarpaneele erzeugten Energie sind eine Reihe von Verlusten zu beachten. Eine einzelne Solarzelle besitzt im bestmöglichen Fall einen Wirkungsgrad von 28,3 %, so dass sie unter optimalen Bedingungen bei einer mittleren Sonnenleistung in Erdnähe von  $p_{\odot} \approx 1369 \frac{\text{W}}{\text{m}^2}$  gut 1 W und ein Paneel somit gut 14 W liefern kann. Aufgrund der Elliptizität des Erdbits ändert sich der Abstand der Erde zur Sonne und damit der Wert der Sonneneinstrahlung jedoch im Verlauf eines Jahres. Für die gegenwärtige Zeit wird in [11] folgende Näherungsgleichung angegeben

$$p_{\odot} = 1371 \frac{\text{W}}{\text{m}^2} \cdot \left( 1 + 0,033 \cdot \cos \left( 360^{\circ} \cdot \frac{n_{\text{Tag}}}{365 \text{ d}} \right) \right) \quad (4.2)$$

wobei  $n_{\text{Tag}}$  die Anzahl der Tage nach dem 3. Januar angibt, an dem sich die Erde im Mittel am Perihel befindet. Damit lässt sich die Strahlungsleistung der Sonne an jedem Tag im Jahr berechnen. In der Realität wird die erzeugte elektrische Leistung der Solarpaneele durch weitere Verluste durch die Richtung und Intensität der Sonneneinstrahlung, sich ändernde Temperaturen und Degradation verringert, die im Folgenden beschrieben werden.

#### 4.3.1 Einflüsse der Sonneneinstrahlung

Der wichtigste Einfluss der Sonneneinstrahlung auf die in einem Solarpaneel erzeugte Leistung ist der Einfallswinkel. Der Winkel  $\gamma$  wird definiert als die Abweichung der Einstrahlungsrichtung von der Normalen der Paneeloberfläche. Er ist also  $0^{\circ}$ , wenn das Sonnenlicht senkrecht einfällt und maximal  $90^{\circ}$ , wenn das Sonnenlicht parallel zur Oberfläche einfällt. Aus der Anschauung ergibt sich, dass die effektiv angestrahlte Fläche, also die aus Richtung der Sonnenstrahlung sichtbare Fläche, mit steigendem  $\gamma$  abnimmt. Zur Berechnung der erzeugten Energie wird ein Faktor  $F_{\text{incid}}$  eingeführt, der die Verringerung der erzeugten Energie aufgrund des Einfallswinkels angibt. In erster Näherung ergibt sich aus der Anschauung  $F_{\text{incid}} = \cos \gamma$ . Dieser Wert gilt sehr gut für  $\gamma < 50^{\circ}$ . Bei größeren Einfallswinkeln ist dieser Zusammenhang jedoch nicht mehr streng gültig, vor allem aufgrund von verstärkter Reflexion am Deckglas der Solarzelle. Darüber hinaus spielen jedoch auch eine verstärkte Absorption und zelleninterne Abschattungen eine Rolle. Streng genommen wirkt sich der Einfallswinkel dabei auf Kurzschlussstrom und Leerlaufspannung unterschiedlich aus, so dass auch eine Temperaturabhängigkeit hinzukommt [12]. Hier wird jedoch eine vereinfachte empirische Gleichung aus [13] zur Beschreibung der Verringerung der erzeugten Energie bei hohen Einfallswinkeln herangezogen, so dass sich für den Verlustfaktor  $F_{\text{incid}}$  ergibt:

$$F_{\text{incid}} = \begin{cases} \cos \gamma, & \gamma \leq 50^{\circ} \\ -0,369 \cdot \cos^3 \gamma + 0,637 \cdot \cos^2 \gamma + 0,75 \cdot \cos \gamma - 0,015, & 50^{\circ} < \gamma < 88,8^{\circ} \\ 0, & \gamma \geq 88,8^{\circ} \end{cases} \quad (4.3)$$

Wie bereits erwähnt gilt das Kosinusetz bis zu einem Einfallswinkel von  $50^\circ$  sehr gut. Für größere Winkel wird die genannte empirische Gleichung verwendet, die allerdings für sehr große Winkel kleiner als null wird. Für diese Winkel wird  $F_{incid}$  daher auf null gesetzt.

Ein weiterer zu berücksichtigender Einfluss ist der Schattenwurf von Instrumenten an Bord, wie zum Beispiel Antennen. Bei älteren Solarpaneelen bedeutete eine Abschattung einer einzelnen Zelle einen Ausfall eines gesamten Strings, da die im Schatten liegende Zelle einen sehr hohen elektrischen Widerstand besitzt. Bei den auf CLAVIS verwendeten Solarpaneelen wird dies durch sogenannte „Bypass-Dioden“ verhindert [14], die die elektrische Leitfähigkeit im Falle einer Abschattung sicherstellen. Daher wird hier angenommen, dass die erzeugte Energie eines Solarpaneels proportional zur angestrahlten Fläche ist. Diese angestrahlte Fläche  $A_{ill}$  kann unter Berücksichtigung der Abschattung durch an Bord befindliche Teile in STK bestimmt werden.

Befindet sich der Satellit darüber hinaus im Halbschatten, so ist nur ein Teil der Sonnenscheibe für die Solarpaneele sichtbar. Zur Berechnung der erzeugten Leistung kann dieser Anteil  $b_\odot$  in STK ausgegeben werden. Damit ergibt sich die gesamte erzeugte Leistung  $P$  in einem Paneel zu

$$P = p_\odot \cdot A_{ill} \cdot b_\odot \cdot F_{incid} \cdot \eta_{ges} \quad (4.4)$$

wobei  $\eta_{ges}$  den Wirkungsgrad der Solarzellen darstellt, der im Folgenden näher beschrieben wird.

### 4.3.2 Degradation

Verschiedene Einflüsse bewirken eine Abnahme des Wirkungsgrades der Solarzellen mit der Zeit. Man spricht zu Beginn der Mission von „Beginning of Life“ (BOL) und am nominellen Ende der Mission von „End of Life“ (EOL). Für die Bestimmung der zur Verfügung stehenden Energie wird dabei stets der EOL-Wirkungsgrad als Worst-Case-Szenario herangezogen.

Der wichtigste dieser Einflüsse ist die Strahlung, die die erzeugte Leistung mit der Zeit verringert. Für die auf CLAVIS verwendeten Solarpaneele werden Richtwerte vom Hersteller angegeben, die von der Fluenz von Elektronen der Energie 1 MeV abhängen. Mit dem von der ESA online frei zur Verfügung gestelltem Tool SPENVIS (Space Environment Information System) [15] lässt sich die Strahlungsbelastung über die Dauer der Mission für den entsprechenden Orbit bestimmen. Als Ausgabe kann die Fluenz von Elektronen und Protonen verschiedener Energien für die von SPENTROLAB hergestellten Solarpaneele auf eine äquivalente Fluenz von 1 MeV-Elektronen umgerechnet werden. Für die AISat-Mission ergibt sich unter Berücksichtigung der Schutzglasdicke von  $125\mu\text{m}$  eine Fluenz von ungefähr  $1,2 \cdot 10^{13} \text{ cm}^{-2}$  1 MeV-Elektronen über ein Jahr. Die Eingabeparameter für die Berechnung mit SPENVIS finden sich im Anhang B. Diese Fluenz ist um etwa eine Größenordnung niedriger, als die geringste im Datenblatt der Solarzellen angegebene, bei der sich die erzeugte Leistung um 7 % verringert [14]. Für die weitere Analyse wird hier eine Verringerung

der erzeugten Leistung von 3 % nach einem Jahr angenommen, was im Vergleich zu den in [16] als Richtwert für Triple Junction Solarzellen angegebenen 0,5 % im Jahr eine konservative Annahme darstellt. Der entsprechende zu berücksichtigende Wirkungsgrad ist also  $\eta_{rad,EOL} = 0,97$ .

Weitere Einflüsse existieren, die die Leistung der Solarpaneele mit der Zeit reduzieren. So gibt es stets Weltraummüll- oder Mikrometeoritenpartikel, die auf die Paneele einschlagen. Für die damit verbundene Reduktion des Wirkungsgrades der Solarpaneele wird in [11] ein Wert von  $\eta_{deb,EOL} = 0,9975$  nach einem Jahr angegeben. Darüber hinaus können auch die ständig wechselnde Temperatur sowie Ausgasungsvorgänge eine Rolle spielen. Da für die Einflüsse der Temperatur jedoch im Folgenden ein Worst-Case-Szenario angenommen wird, wird dieser Einfluss hier vernachlässigt. Gleiches gilt für Ausgasungsvorgänge, die bei den hier verwendeten Solarzellen unerheblich sind.

### 4.3.3 Einflüsse der Temperatur

Die Richtwerte der Solarzelle werden zumeist unter Standardbedingungen bei einer Temperatur von 28 °C angegeben. Ist die eigentliche Temperatur der Paneele größer, erhöhen sich jedoch der Kurzschlussstrom und die Stromstärke am MPP, wobei sich die entsprechenden Spannungswerte verringern. Da die Verringerung der Spannungen größer ist als die Erhöhung des Stroms, ergibt sich insgesamt eine Verringerung der Leistung einer Solarzelle bei steigender Temperatur. Der Effekt ist auch in Abbildung 4.2 gezeigt. Verringert sich die Temperatur der Paneele, so ergibt sich eine höhere Leistung.

Die Temperatur der Solarpaneele zu jedem beliebigen Zeitpunkt einer Mission zu bestimmen ist nicht trivial. Wie in Abschnitt 3.1 ausgeführt, schwankt die Temperatur des Satelliten und insbesondere der direkt angestrahlten Solarpaneele während eines Orbits deutlich. Sie ist am Ende der längsten Schattenzeit am niedrigsten, wenn der Satellit eine große Menge an Wärme abgestrahlt hat und kann dann unter 0 °C liegen. Nach einer längeren Zeit ohne Eklipsen kann sich ein Gleichgewicht einstellen, bei dem die Maximaltemperatur des Satelliten erreicht ist. Diese kann über 100 °C betragen. Die Temperatur der einzelnen Paneele hängt dabei jedoch auch stark von der jeweiligen Orientierung zur Sonne ab.

Im Datenblatt der verwendeten Solarpaneele [14] finden sich die MPP-Werte für den Strom bei 28 °C (Index 0),  $I_{0,mpp} = 433,91 \text{ mA}$ , sowie die Spannung bei 28 °C,  $U_{0,mpp} = 2,350 \text{ V}$ . Daraus ergibt sich die am MPP erzeugte Leistung einer Solarzelle bei 28 °C zu

$$P_{0,mpp} = I_{0,mpp} \cdot U_{0,mpp} = 433,91 \text{ mA} \cdot 2,350 \text{ V} = 1,020 \text{ W}$$

Darüber hinaus werden Temperaturkoeffizienten für diese Werte angegeben, die für den Temperaturbereich von 15 °C bis 80 °C gültig sind und wiederum abhängig von der Bestrahlung sind. Da die niedrigste angegebene Fluenz jedoch nicht erreicht wird (siehe Abschnitt 4.3.2), werden hier die BOL-Werte verwendet. Diese sind für den Strom  $\frac{dI_{mpp}}{dT} = 0,0692 \frac{\text{mA}}{^\circ\text{C}}$  und für die Spannung  $\frac{dU_{mpp}}{dT} = -6,7 \frac{\text{mV}}{^\circ\text{C}}$ .

In [16] wird  $67^\circ\text{C}$  als Richtwert für die Temperatur eines senkrecht angestrahlten Solarpanels für Satelliten auf LEO angegeben. Zur Vereinfachung der Berechnung der erzeugten Leistung wird hier die Temperatur der Solarpaneele im Sinne einer konservativen Auslegung als konstant  $80^\circ\text{C}$  angenommen. Strom, Spannung und erzeugte Leistung am MPP ergeben sich bei dieser Temperatur zu

$$\begin{aligned} I_{mpp}(80^\circ\text{C}) &= I_{0,mpp} + \frac{dI_{mpp}}{dT} \cdot \Delta T = 433,91 \text{ mA} + 0,0692 \frac{\text{mA}}{^\circ\text{C}} \cdot 52^\circ\text{C} = 437,51 \text{ mA} \\ U_{mpp}(80^\circ\text{C}) &= U_{0,mpp} + \frac{dU_{mpp}}{dT} \cdot \Delta T = 2,350 \text{ V} - 6,7 \frac{\text{mV}}{^\circ\text{C}} \cdot 52^\circ\text{C} = 2,002 \text{ V} \\ \Rightarrow P_{mpp}(80^\circ\text{C}) &= I_{mpp}(80^\circ\text{C}) \cdot U_{mpp}(80^\circ\text{C}) = 437,51 \text{ mA} \cdot 2,002 \text{ V} = 0,875 \text{ W} \end{aligned}$$

Daraus ergibt sich der Wirkungsgrad zur Berücksichtigung der Temperaturabhängigkeit  $\eta_{temp}$  zu

$$\eta_{temp} = \frac{P_{mpp}(80^\circ\text{C})}{P_{0,mpp}} = \frac{0,875 \text{ W}}{1,020 \text{ W}} = 0,858$$

Dieser Wert stimmt ungefähr mit dem in [16] angegebenen Richtwert überein.

#### 4.3.4 Sonstige Einflüsse

Weitere Verluste ergeben sich in einem Array von Solarzellen aufgrund der Verschaltung der einzelnen Zellen. Verbindungselemente zwischen den einzelnen Zellen sowie die Bypass-Dioden erhöhen den Innenwiderstand und reduzieren die effektiv nutzbare Solarzellenfläche. Außerdem können diese bei großen Einstrahlwinkeln auch eine gewisse Abschattung verursachen. Ein Richtwert für diese Verluste wird in [16] mit  $\eta_{ass} = 0,85$  angegeben. Darüber hinaus verursacht das Deckglas einer Solarzelle einen gewissen Verlust, da kein Material perfekt durchlässig für alle ausnutzbaren Wellenlängen ist. Ein entsprechender Richtwert wird in [11] mit  $\eta_{cov} = 0,99$  angegeben. Außerdem ergeben sich Abweichungen der tatsächlichen Strom- und Spannungswerte von den theoretisch optimalen, so dass der MPP nicht genau getroffen wird. Da diese Verluste jedoch genau genommen nicht von den Solarpaneelen, sondern vom DEPS verursacht werden, werden sie im folgenden Abschnitt berücksichtigt.

Der Gesamtwirkungsgrad eines Solarpanels in Gleichung (4.4) ergibt sich dann aus den einzelnen beschriebenen Verlusten unter Berücksichtigung des nominellen Wirkungsgrades  $\eta_{nom} = 28,3\%$  zu

$$\begin{aligned} \eta_{ges} &= \eta_{nom} \cdot \eta_{rad,EOL} \cdot \eta_{deb,EOL} \cdot \eta_{temp} \cdot \eta_{ass} \cdot \eta_{cov} \\ &= 0,283 \cdot 0,97 \cdot 0,9975 \cdot 0,858 \cdot 0,85 \cdot 0,99 = 0,1977 \hat{=} 19,77\% \end{aligned}$$

### 4.4 Auf- und Entladung der Batterie

Zur Betrachtung der Auf- und Entladung der Batterie ist zunächst deren Kapazität von Interesse. In der AISat-Konfiguration des CLAVIS gibt es zwei parallel geschaltete

Batterien, deren Kapazität von der Temperatur sowie der Entladerate abhängt, wobei die Kapazität mit steigender Temperatur und sinkender Entladerate steigt. Tabellierte Werte dazu finden sich in [9]. Die Entladeraten werden oft als Vielfaches der Kapazität in Amperestunden  $C_{batt}^*$  angegeben, wobei damit eine Auf- bzw. Entladegeschwindigkeit gemeint ist, die die gesamte Kapazität in einer Stunde wiederherstellen kann. Die Raten besitzen also eine Stromstärke als Einheit. Geht man von einer Batterietemperatur von  $0^\circ\text{C}$  aus, beträgt die Kapazität der gesamten Batterie bei einer Entladegeschwindigkeit von  $C_{batt}^*/5$  ca.  $C_{batt}^* = 4,64 \text{ Ah}$ , also  $2,32 \text{ Ah}$  pro einzelner Batterie. Die Temperatur von  $0^\circ\text{C}$  ist dabei eine Worst-Case-Annahme, da der Heizer stets für eine höhere Betriebstemperatur sorgt. Wie im Folgenden gezeigt wird, stellt eine Entladerate von  $C_{batt}^*/5$  ebenfalls eine konservative Annahme dar. Sowohl die Ent- als auch die Aufladerate sollten zu keinem Zeitpunkt einen Wert von  $1 \cdot C_{batt}^*$  überschreiten [9].

Die von der Batterie zur Verfügung gestellte Spannung variiert mit der restlichen in der Batterie befindlichen Kapazität. In [17] wird ein nomineller Wert für eine Zelle von  $3,7 \text{ V}$  angegeben. Da eine Batterie aus zwei in Reihe geschalteten Zellen besteht, ist die nominelle Spannung der gesamten Batterie damit  $U_{nom} = 7,4 \text{ V}$ . Wie aus der Entladecharakteristik in [9] hervorgeht, liegt dieser Wert unter der Batteriespannung bei der maximalen DoD von  $20\%$  und ist somit eine konservative Annahme. Damit ergibt sich die Kapazität der Batterie in Wattstunden  $C_{batt}$  zu

$$C_{batt} = C_{batt}^* \cdot U_{nom} = 4,64 \text{ Ah} \cdot 7,4 \text{ V} = 34,34 \text{ Wh}$$

Ist die Dauer der Eklipse  $t_{ecl}$  in einem Orbit bekannt, kann über die bekannte Umlaufdauer  $t_U$  (siehe Gleichung (3.2)) die Zeit im Sonnenlicht zu  $t_{ill} = t_U - t_{ecl}$  berechnet werden. Die während der Zeit im Erdschatten benötigte Energie  $E_{ecl}$  ergibt sich dann aus der maximalen aufgenommenen Leistung  $P_{auf} = 4,29 \text{ W}$  (siehe Gleichung (4.1)) und den Verlusten der Energieübertragung  $\eta_{bus} = 0,95$  zu

$$E_{ecl} = \frac{P_{auf} \cdot t_{ecl}}{\eta_{bus}} \quad (4.5)$$

Der Wirkungsgrad  $\eta_{bus}$  ist dabei der schlechteste Fall für die Energieübertragung im  $3,3 \text{ V}$ - bzw. im  $5 \text{ V}$ -Bus [8]. Die Verluste im  $12 \text{ V}$ -Bus sind etwas größer. Da dieser aber nur für einige wenige Systeme verwendet wird, kann  $0,95$  als Richtwert angenommen werden. Weitere Verluste gibt es hier nicht, da die Energieversorgung während einer Schattenphase einzig von der Batterie übernommen wird. Die Energieübertragung zwischen Batterie und Bus läuft praktisch verlustfrei.

Während der Zeit in der Sonne muss genug Energie von den Solarpaneelen erzeugt werden, um alle Systeme zu versorgen und die Batterie wieder aufzuladen. Die während dieser Zeit für den Betrieb der Satellitensysteme und der Nutzlast benötigte Energie  $E_{ill}$  ergibt sich analog zu der in der Schattenphase benötigten Energie. Hier muss allerdings zusätzlich der Wirkungsgrad der BCRs  $\eta_{BCR} = 0,9$  [8] berücksichtigt werden, da die Energieübertragung zwischen den Paneelen und dem Bus immer über die BCRs läuft. Es ist

$$E_{ill} = \frac{P_{auf} \cdot t_{ill}}{\eta_{bus} \cdot \eta_{BCR}} \quad (4.6)$$

Ist die während der Sonnenphase durch die Solarpaneele erzeugte Energie  $E_{pan}$  aus Simulationsrechnungen unter Berücksichtigung der in Abschnitt 4.3 beschriebenen Verluste bekannt, kann die zur Aufladung der Batterie verbleibende Energie  $E_{cha}$  berechnet werden. Dazu muss neben dem Wirkungsgrad der BCRs auch der Battery Return Factor  $RF = 1,1$  (siehe Abschnitt 4.1) berücksichtigt werden. Beide Verluste bedeuten im Grunde eine Erhöhung der benötigten Energie zur Aufladung der Batterie oder im Umkehrschluss, wie hier verwendet, eine Verringerung der zur Aufladung verfügbaren Energie. Damit ergibt sich  $E_{cha}$  zu

$$E_{cha} = (E_{pan} - E_{ill}) \cdot \frac{\eta_{BCR}}{RF} \quad (4.7)$$

Vergleicht man diesen Wert nun mit der während der Eklipse benötigten Energie, kann man sagen, dass genug Energie während der Zeit in der Sonne erzeugt wird, wenn  $E_{cha} > E_{ecl}$ . Die vorhandene Marge  $m$  kann dann in Prozent ausgedrückt werden als

$$m = \frac{E_{cha} - E_{ecl}}{E_{cha}} \cdot 100\% \quad (4.8)$$

Ist dieser Wert negativ, so wird nicht genug Energie erzeugt. Darüber hinaus kann die tatsächliche DoD  $DoD_{act}$  in Prozent unter Berücksichtigung der Kapazität der Batterie bestimmt werden zu

$$DoD_{act} = \frac{E_{ecl}}{C_{batt}} \cdot 100\% \quad (4.9)$$

Dieser Wert sollte wie oben erwähnt 20 % nicht überschreiten. Ist dies der Fall, ist auch die Annahme, dass die nominelle Spannung der Spannung bei einer DoD von 20 % entspricht, nicht mehr gültig, so dass die gesamte Rechnung hinfällig wird.

Da sowohl die Auf- als auch die Entladerate der Batterie begrenzt sind [9], sollten auch diese berechnet werden, um zu überprüfen, ob der Batterie die benötigten Energien in der entsprechenden Zeit zugeführt bzw. entzogen werden können. Die Auf- und Entladeraten  $I_{cha}$  und  $I_{dis}$  ergeben sich aus der während der Eklipse benötigten Energie unter Berücksichtigung der nominellen Batteriespannung sowie des Battery Return Factors und der Verluste beim Aufladen zu

$$I_{cha} = \frac{E_{ecl} \cdot RF}{U_{nom} \cdot t_{ill} \cdot \eta_{BCR}} \quad (4.10)$$

$$I_{dis} = \frac{E_{ecl}}{U_{nom} \cdot t_{ecl}} \quad (4.11)$$

Setzt man nun Gleichung (4.5) in Gleichung (4.11) ein, kürzt sich die Dauer der Eklipse heraus, so dass nur noch hier als konstant angenommene Werte übrig bleiben. Es ist

$$I_{dis} = \frac{P_{auf} \cdot t_{ecl}}{\eta_{bus} \cdot U_{nom} \cdot t_{ecl}} = \frac{P_{auf}}{\eta_{bus} \cdot U_{nom}} = \frac{4,29 \text{ W}}{0,95 \cdot 7,4 \text{ V}} = 0,61 \text{ A}$$

Da hier eine konstante Leistungsaufnahme des Satellitensystems angenommen wird, ist die Entladerate der Batterie wenn keine Energie von den Solarpaneelen zur Verfügung steht folglich ebenfalls konstant. Wie oben bereits erwähnt, wird diese Rate oft



als Vielfaches der Batteriekapazität in Amperestunden  $C_{batt}^*$  angegeben. Damit ist

$$I_{dis}^* = \frac{I_{dis}}{C_{batt}^*} = \frac{0,61 \text{ A}}{4,64 \text{ Ah}} \hat{=} 0,13 \cdot C_{batt}^*$$

Die in der Gebrauchsanweisung der Batterie angegebene maximale Entladerate von  $I_{dis}^* = 1 \cdot C_{batt}^*$  wird also nicht erreicht. Darüber hinaus zeigt sich, dass die oben angenommene Entladerate von  $C_{batt}^*/5$  ebenfalls unterschritten wird. Allerdings wird bei dieser Berechnung die Kapazität verwendet, die aus der ursprünglichen Annahme, dass die Entladerate kleiner als  $C_{batt}^*/5$  ist, abgeleitet wurde (siehe Anfang dieses Abschnitts). Somit wird diese Annahme unter Verwendung der selben Annahme verifiziert. Dies liegt jedoch an der Angabe der Batteriekapazität als Funktion der Entladerate und kann hier nicht vermieden werden. Grundsätzlich ist aber davon auszugehen, dass die Batterie bei Betriebsbeginn eine verhältnismäßig hohe Kapazität besitzt, so dass die tatsächliche Entladerate noch unter der hier angenommenen liegt.

Mit den hier dargestellten Berechnungen lässt sich unter Verwendung der mit Hilfe von STK bestimmbarren Ausgaben berechnen, ob das Energieversorgungssystem zu jedem Zeitpunkt einer Mission genug Energie für alle Systeme zur Verfügung stellen kann. Für eine längere Mission können die maximale und die durchschnittliche Eklipsendauer zur Berechnung herangezogen werden, so dass eine Aussage zur Gesamtmission getroffen werden kann.

Für die Betrachtung zukünftiger CLAVIS-Missionen ist die Leistungsaufnahme der Satellitensysteme nicht im Detail bekannt. In diesem Fall kann die maximal mögliche Leistungsaufnahme  $P_{auf}$  unter Berücksichtigung der bekannten erzeugten Energie  $E_{pan}$  bestimmt werden, so dass die Batterie während einer Sonnenlichtperiode gerade voll geladen wird. Daher wird

$$m \stackrel{!}{=} 0$$

gefordert, so dass sich nach Gleichung (4.8)  $E_{cha} = E_{ecl}$  ergibt. Ersetzt man also in Gleichung (4.7)  $E_{cha}$  durch  $E_{ecl}$  und setzt entsprechend Gleichungen (4.5) und (4.6) ein, ergibt sich

$$\frac{P_{auf} \cdot t_{ecl}}{\eta_{bus}} = E_{pan} \cdot \frac{\eta_{BCR}}{RF} - \frac{P_{auf} \cdot t_{ill}}{\eta_{bus} \cdot \eta_{BCR}} \cdot \frac{\eta_{BCR}}{RF}$$

In dieser Gleichung kommen nun noch die Verlustfaktoren des auf der AISat-Konfiguration verwendeten DEPS vor. Da aber zukünftige Versionen des CLAVIS voraussichtlich ein ähnliches DEPS besitzen werden, werden diese Faktoren beibehalten. Nach einigen Umformungen ergibt sich für die gesuchte Leistungsaufnahme  $P_{auf}$

$$P_{auf} = \frac{E_{pan} \cdot \eta_{BCR} \cdot \eta_{bus}}{RF \cdot t_{ecl} + t_{ill}} \quad (4.12)$$

Mit Hilfe dieser Gleichung kann also ein Richtwert für die maximal mögliche Leistungsaufnahme bestimmt werden. Es kann allerdings keine Aussage zur maximalen DoD oder der Laderate getroffen werden, da die Batteriekapazität nicht bekannt ist. Für den Einzelfall müssen diese Beschränkungen berücksichtigt werden.

# Kapitel 5

## Kommunikationssystem

### 5.1 Gesamtsystem

Bei der Betrachtung des Raumsegments des Kommunikationssystems von CLAVIS muss zwischen der schon im Detail definierten AISat-Konfiguration und der Standardkonfiguration unterschieden werden. Auf der AISat-Plattform ist lediglich ein Transceiver vorgesehen, der eine Kommunikation mit Funkwellen im 70-Zentimeter-Band des Amateurfunks ermöglicht. Für die Standard-Plattform ist darüber hinaus ein S-Band-Transceiver geplant, der im Frequenzbereich zwischen 2,7 GHz und 3,5 GHz operiert. Da jedoch noch zu wenig Parameter dieses Systems bekannt sind, um fundierte Aussagen über die Signalqualität und andere Kennzahlen treffen zu können, wird hier nur das Kommunikationssystem der AISat-Konfiguration betrachtet.

Das Amateurfunkfrequenzband, das für Satellitenkommunikation vorgesehen ist, befindet sich zwischen 435 MHz und 438 MHz im „Ultra High Frequency“ (UHF)-Bereich. Der UHF-Transceiver an Bord von CLAVIS wurde von der Firma GOMSPACE hergestellt und kann Funkwellen im Bereich von 435 MHz bis 437 MHz aussenden und empfangen. Zum Senden von Daten an eine Bodenstation („Downlink“) werden die zu übertragenden Daten im Modem zunächst durch eine digitale BSK-Modulation („Binary Shift Keying“) auf eine Welle aufmoduliert und unter Verwendung eines sogenannten (255,223)-Reed-Solomon-Codes kanalkodiert. Im eigentlichen Sender wird dann eine analoge Frequenzmodulation verwendet, um die Daten auf die endgültige Trägerwelle aufzumodulieren. Diese Welle wird daraufhin mit einer Leistung von 2 W zu den Antennen übertragen. Zwischen Sender und den Antennen befindet sich ein sogenannter 3 dB-Richtkoppler („3 dB-Coupler“), der das Signal in zwei gleichstarke Signale aufteilt, so dass ein Signal mit einer Leistung von ca. 1 W an beiden Antennen ankommt. Zum Senden und Empfangen der Signale gibt es zwei Monopolantennen, von denen sich eine innerhalb der Helixantenne befindet und die andere auf der Grundplatte angebracht ist und im 45°-Winkel nach außen geneigt ist.

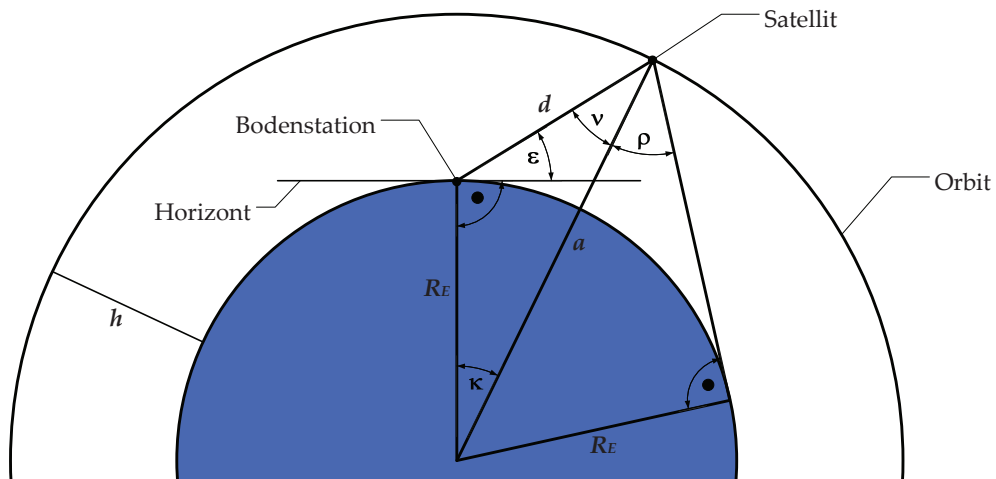
Als Bodenstation für die AISat- und die Folgemissionen soll zum einen die am Institut für Raumfahrtssysteme des DLR in Bremen vorhandene Empfangsanlage mit einer zirkularen Helixantenne und weiteren Komponenten aufgerüstet werden. Ihre Position in geodetischen Koordinaten ist bei 53,111° Nord und 8,862° Ost. Darüber hinaus befindet sich eine Bodenstation des DLR in Inuvik in Kanada im Aufbau. Sie

liegt bei  $68,318^\circ$  Nord und  $133,545^\circ$  West und ist aufgrund ihrer Lage nördlich des Polarkreises besonders geeignet, Signale von Satelliten auf polnahen Orbits wie zum Beispiel sonnensynchronen Orbits über lange Zeiten zu empfangen.

In der Empfangsanlage der Bodenstationen wird das Signal zunächst vorverstärkt. Daraufhin wird zuerst die analoge Funkwelle demoduliert und dann das digital modulierte Signal. Erst danach können die gesendeten Bits aufbereitet und analysiert werden. Bei beiden Demodulationen muss das Signal deutlich stärker sein als das unvermeidliche Rauschen, damit die Fehlerfreiheit der Daten gewährleistet ist (siehe Abschnitt 5.3).

## 5.2 Geometrie der Funkverbindung

Zur Betrachtung der Funkverbindung zwischen einem Satelliten und einer Bodenstation ist zunächst eine geometrische Betrachtung sinnvoll, um die während einer Kontaktperiode übertragbare Datenmenge abschätzen zu können. Weitere Ausführungen dazu finden sich in [16]. Abbildung 5.1 zeigt diese Geometrie vereinfacht für einen Satelliten in einem kreisförmigen LEO zu einem beliebigen Zeitpunkt an dem eine Sichtverbindung besteht. Darin ist  $d$  der Abstand zwischen Satellit und Bodenstation,  $\rho$  der halbe Winkeldurchmesser der Erde vom Satelliten aus gesehen,  $\varepsilon$  die Elevation des Satelliten von der Bodenstation aus gesehen,  $\kappa$  der Winkel zwischen Satellit und Bodenstation vom Erdmittelpunkt aus gesehen und  $\nu$  der Winkel zwischen Nadir und Bodenstation vom Satelliten aus gesehen;  $\nu$  wird auch als „Nadirwinkel“ bezeichnet.



**Abbildung 5.1:** Geometrie der Funkverbindung zwischen einem Satelliten in einem kreisförmigen, erdnahen Orbit und einer Bodenstation

Für den Winkel  $\rho$  ergibt sich, wie schon in Abschnitt 3.2 für den Öffnungswinkel  $\alpha$  beschrieben:

$$\sin \rho = \frac{R_E}{R_E + h} = \frac{R_E}{a} \quad (5.1)$$

Kennt man also die Bahnhöhe des Satelliten und die Elevation, so kann man den Nadirwinkel  $\nu$  wie folgt berechnen. Aus dem linken Dreieck in Abbildung 5.1 ergibt sich nach dem Sinussatz

$$\sin \nu = \frac{R_E}{a} \cdot \sin(\varepsilon + 90^\circ) \quad (5.2)$$

Berücksichtigt man nun, dass  $\sin(\varepsilon + 90^\circ) = \cos \varepsilon$  ist, ergibt sich unter Verwendung von Gleichung (5.1)

$$\sin \nu = \sin \rho \cdot \cos \varepsilon \quad (5.3)$$

Damit ergibt sich  $\kappa$  aus der Winkelsumme im linken Dreieck zu

$$\kappa = 180^\circ - (90^\circ + \varepsilon) - \nu = 90^\circ - \varepsilon - \nu \quad (5.4)$$

Darüber hinaus lässt sich ebenfalls unter Verwendung des Sinussatzes der Abstand zwischen Satellit und Bodenstation  $d$  bestimmen zu

$$d = R_E \cdot \frac{\sin \kappa}{\sin \nu} \quad (5.5)$$

Für die folgenden Betrachtungen ist es sinnvoll, die minimale und die maximale Elevation des Satelliten während einer Funkverbindung zu definieren. Bei zu geringen Elevationen ist der Weg des Funksignals durch die niedrigen, besonders dichten und turbulenten Atmosphärenschichten so lang, dass es zu stark gedämpft wird, als dass eine effektive Datenübertragung aufrecht erhalten werden könnte. Die minimale Elevation, bei der eine Funkverbindung gerade noch möglich ist, liegt meist zwischen  $5^\circ$  und  $10^\circ$  und wird hier zu  $\varepsilon_{min} = 10^\circ$  gesetzt. Die maximale Elevation während eines Überflugs  $\varepsilon_{max}$  kann mit Hilfe von STK bestimmt werden. Zur Berechnung der übertragbaren Datenmengen werden nun die entsprechenden  $\kappa_{max}$  und  $\kappa_{min}$  herangezogen, die sich mit Hilfe der Gleichungen (5.1), (5.3) und (5.4) ergeben zu

$$\kappa_{max} = 90^\circ - \varepsilon_{min} - \arcsin\left(\frac{R_E}{a} \cdot \cos \varepsilon_{min}\right) \quad (5.6)$$

$$\kappa_{min} = 90^\circ - \varepsilon_{max} - \arcsin\left(\frac{R_E}{a} \cdot \cos \varepsilon_{max}\right) \quad (5.7)$$

Die theoretische maximale Zeit für den Bodenkontakt  $t_{max}$  lässt sich dann mit Hilfe der Umlaufzeit aus Gleichung (3.2) berechnen zu

$$t_{max} = t_U \cdot \frac{2 \cdot \kappa_{max}}{360^\circ} = t_U \cdot \frac{\kappa_{max}}{180^\circ} \quad (5.8)$$

Dabei wird allerdings der optimale Fall aus Abbildung 5.1 vorausgesetzt, bei dem sich der Satellit zu einem Zeitpunkt im Zenit der Bodenstation befindet, so dass  $\kappa_{min} = 0^\circ$  ist. Für einen beliebigen Fall ergibt sich der Bruchteil der tatsächlich zur Verfügung stehenden Bodenkontaktzeit  $F_{acc}$  nach [16] zu

$$F_{acc} = \frac{1}{\kappa_{max}} \cdot \arccos\left(\frac{\cos \kappa_{max}}{\cos \kappa_{min}}\right) \quad (5.9)$$

Dies führt offensichtlich nur dann zu einem sinnvollen Ergebnis, wenn  $\kappa_{\max} > \kappa_{\min}$ , also  $\varepsilon_{\max} > \varepsilon_{\min}$  ist; anderenfalls wird das Argument der Arcuskosinusfunktion größer als eins. Damit berechnet sich die übertragbare Datenmenge  $D$  unter Berücksichtigung der Datenrate  $R_{\text{dat}}$  und der zur Initialisierung der Funkverbindung nötigen Zeit  $t_{\text{init}}$  zu

$$D = R_{\text{dat}} \cdot (F_{\text{acc}} \cdot t_{\text{max}} - t_{\text{init}}) \cdot F_{\text{red}} \quad (5.10)$$

wobei mit  $F_{\text{red}}$  mögliche Reduktionen der Datenmenge berücksichtigt werden. Diese können bedingt sein durch die Auslastung der Bodenstation durch andere Satelliten, durch eventuelle Ausfallzeiten zu denen die Bodenstation nicht besetzt oder nicht erreichbar ist oder ähnliches. Der Term  $F_{\text{acc}} \cdot t_{\text{max}}$  ist dabei die Bodenkontaktzeit, die in STK auch direkt bestimmt werden kann. Auf CLAVIS wird zunächst eine Datenrate von 4800 Bits pro Sekunde verwendet, wobei diese je nach Qualität des Signalempfangs auch während der Mission angepasst werden kann.

## 5.3 Qualität der Funkverbindung

### 5.3.1 Grundlegende Betrachtungen

Neben der übertragbaren Datenmenge spielt auch die Qualität der Datenübertragung eine wichtige Rolle. Als Grundlage der Betrachtung dient dabei das Linkbudget, das für die AISat-Mission erstellt wurde [18]. Darin zeigt sich, dass bei der Datenübertragung von der Bodenstation zum Satelliten auch im schlechtesten Fall bei einer Elevation von  $10^\circ$  ausreichende Spielräume vorhanden sind. Dieser sogenannte „Uplink“ wird hier deshalb nicht weiter betrachtet. Bei der Datenübertragung vom Satellit zur Bodenstation, dem sogenannten „Downlink“ ist dies jedoch nicht der Fall, weshalb eine detaillierte Simulation durchgeführt wird.

Der wichtigste Kennwert für die Übertragungsqualität ist das Signal-zu-Rauschverhältnis. Dieses wird jedoch unterschiedlich definiert, je nachdem, ob eine digitale oder analoge Modulation vorliegt. Bei einer digitalen Modulation ist das Verhältnis der pro Bit empfangenen Energie zu der spektralen Rauschleistungsdichte der entscheidende Wert, der die Fehlerhäufigkeit, die sogenannte „Bit Error Rate“, bestimmt. Aus dem Linkbudget für die AISat-Konfiguration geht jedoch hervor, dass dieses Verhältnis bei der Demodulation des digitalen Signals auch im ungünstigsten Fall deutlich über dem geforderten Mindestwert liegt. Kritischer ist die Situation bei der Demodulation des analogen Signals. Hier ist das Verhältnis der empfangenen Leistung der Trägerwelle (in der englischsprachigen Literatur auch als „Carrier“ bezeichnet)  $P_r$  zu der empfangenen Rauschleistung  $N$  ausschlaggebend („Carrier-to-Noise-Ratio“, CNR). Es zeigt sich, dass dieses Verhältnis im schlechtesten Fall unterhalb des geforderten Schwellenwertes liegt, weshalb dieses Verhältnis im Folgenden als Gütekriterium der Datenübertragung betrachtet wird.

Zur Berechnung des CNR muss zunächst die Sendefrequenz  $f$  des Kanals bekannt sein. Sie liegt bei der AISat-Konfiguration bei ca.  $f = 436 \text{ MHz}$  (siehe Abschnitt 5.1). Die Frequenz lässt sich unter Berücksichtigung der Ausbreitungsgeschwindigkeit einer

elektromagnetischen Welle im Vakuum, der Lichtgeschwindigkeit  $c = 299.792.458 \frac{\text{m}}{\text{s}}$  [16], in die Wellenlänge  $\lambda$  mit Hilfe der bekannten Gleichung

$$\lambda = \frac{c}{f}$$

überführen. Damit ergibt sich eine Wellenlänge von ca.  $\lambda = 0,69 \text{ m}$ . Im Folgenden werden nun die Kenndaten der gesamten Funkverbindung vom Sender über die zu überwindende Strecke bis zum Empfänger genauer betrachtet. Die Betrachtungen sind angelehnt an die Darstellungen in [16] und [19].

### 5.3.2 Gewinne und Verluste der Funkverbindung

Der Sender moduliert die zu übertragenden Daten auf eine elektromagnetische Welle mit einer Sendeleistung  $P_t = 2 \text{ W}$ . Das Signal wird über eine Leitung zu den beiden Antennen mit dem Antennengewinn  $G_t$  übertragen. Der Antennengewinn ist dabei das Verhältnis der in Hauptabstrahlrichtung abgestrahlten Intensität der realen Antenne zu der von einer hypothetischen verlustfreien, in alle Richtungen gleich stark abstrahlenden Antenne (Isotropstrahler) abgestrahlten Intensität. Damit ist der Antennengewinn dimensionslos. Er wird jedoch, wie die meisten anderen Werte in der folgenden Betrachtung, oftmals in Dezibel (dB) angegeben. Beim Antennengewinn wird zu der Dezibelnotation noch ein „i“ hinzugefügt, um anzudeuten, dass er auf einen Isotropstrahler bezogen angegeben wird. Es ist

$$G_t^{dBi} = 10 \cdot \log_{10}(G_t)$$

Der Antennengewinn der Sendeantennen auf CLAVIS ist  $G_t^{dBi} = 3 \text{ dBi}$ . Auf die gleiche Weise können auch dimensionsbehaftete Größen in Dezibel angegeben werden, wobei die entsprechende Einheit mit notiert wird. So gilt zum Beispiel für die Sendeleistung  $P_t$  in Watt

$$P_t^{dBW} = 10 \cdot \log_{10}(P_t) = 10 \cdot \log_{10}(2 \text{ W}) \approx 3 \text{ dBW}$$

In der Leitung zwischen Sender und Antenne entstehen beim Senden der Daten verschiedene Verluste. Zum einen kann ein Signal durch keine reale Leitung komplett verlustfrei übertragen werden, was zu einem Leitungsverlust führt. Dieser kann sofern der Verlust der verwendeten Leitungen bekannt ist, direkt aus der Länge der Leitungen bestimmt werden. Zum anderen treten Verluste durch Reflexion der übertragenen Wellen auf. Beide Verluste werden durch den Verlustkator  $L_{t,line}$  zusammengefasst. Das zu übertragene Signal wird außerdem durch den 3 dB-Richtkoppler gleichmäßig auf die beiden Antennen aufgeteilt, so dass es bei beiden Antennen ungefähr mit der halben Sendeleistung (dies entspricht einer Verringerung um ungefähr 3 dB) ankommt. Da das Bauteil jedoch nicht ideal ist, ist der zu berücksichtigende Verlust  $L_{t,3dB}$  etwas größer als 0,5. Unter Berücksichtigung dieser Verluste lässt sich nun die sogenannte äquivalente isotrope Strahlungsleistung („Equivalent Isotropic Radiated Power“, EIRP) des Satelliten angeben zu

$$EIRP = P_t \cdot G_t \cdot L_{t,line} \cdot L_{t,3dB}$$

Diese Gleichung kann auch für Werte in Dezibel umgeschrieben werden, wobei aus der Multiplikation aufgrund der Logarithmierung eine Addition wird. Es ist

$$EIRP^{dBW} = P_t^{dBW} + G_t^{dBi} + L_{t,line}^{dB} + L_{t,3dB}^{dB} \quad (5.11)$$

Dabei ist zu beachten, dass die Verluste, die vor der Umrechnung in Dezibel im Intervall  $[0...1]$  lagen, nun negativ sind und somit zu einer Verringerung der EIRP führen. Aus dem Linkbudget geht hervor, dass für CLAVIS  $L_{t,line}^{dB} = -0,5 \text{ dB}$  und  $L_{t,3dB}^{dB} = -3,4 \text{ dB}$  sind. Damit ergibt sich

$$EIRP^{dBW} = 3 \text{ dBW} + 3 \text{ dBi} - 0,5 \text{ dB} - 3,4 \text{ dB} = 2,1 \text{ dBW}$$

Auf der Funkstrecke sind weitere Verluste zu berücksichtigen. Der wichtigste Verlust wird dabei durch die sogenannte Freiraumdämpfung hervorgerufen. Die Freiraumdämpfung gibt die Verringerung der Leistungsdichte einer abgestrahlten Welle aufgrund der Ausbreitung im Raum an und hängt neben der Entfernung  $d$  (siehe Abschnitt 5.2) auch von der Frequenz bzw. der Wellenlänge des Signals ab. Der entsprechende zu berücksichtigende Verlust  $L_{fs}$  ergibt sich nach [16] zu

$$L_{fs} = \left( \frac{\lambda}{4\pi d} \right)^2 \quad (5.12)$$

Da die Hauptausstrahlrichtung des Senders auf CLAVIS außerdem praktisch nie exakt auf die Mitte der Empfangsantenne trifft, ergibt sich eine Verringerung des nutzbaren Antennengewinns. Daher wird ein sogenannter „Pointing Loss“ der Sendeantenne  $L_{t,point}$  eingeführt. Im Linkbudget für die AISat-Konfiguration wird ein relativ großer Verlust von  $L_{t,point}^{dB} = -13 \text{ dB}$  angegeben, mit dem auch die maximale Abweichung von der Hauptausstrahlrichtung von  $90^\circ$  berücksichtigt wird. Weitere Verluste ergeben sich durch die unvermeidbare Polarisierung der Welle und werden durch  $L_{pol}^{dB} = -0,5 \text{ dB}$  berücksichtigt. Darüber hinaus spielen, in Abhängigkeit der Frequenz, Verluste in der Atmosphäre  $L_{atm}$  eine Rolle. Bei Frequenzen unterhalb von ca. 600 MHz sorgen Elektronen in der Ionosphäre für Verluste, bei Frequenzen oberhalb von ca. 4 GHz steigt die Absorption durch Sauerstoff und Wasserdampf schnell an. In dem Bereich dazwischen sind Atmosphärenverluste praktisch vernachlässigbar, weshalb dieser Bereich bevorzugt für die Kommunikation mit Satelliten genutzt wird. Für die hier verwendete Frequenz von 436 MHz werden Ionosphärenverluste von  $L_{atm}^{dB} = -0,5 \text{ dB}$  angenommen [18]. Weitere Verluste können sich durch Interferenzen mit Funksignalen auf ähnlichen Frequenzen oder durch fehlerhafte Wellenausbreitung ergeben. Da diese aber schwer abzuschätzen sind, werden sie hier vernachlässigt. Der Gesamtverlust auf der Funkstrecke  $L_{path}$  ergibt sich also zu

$$\begin{aligned} L_{path} &= L_{fs} \cdot L_{t,point} \cdot L_{pol} \cdot L_{atm} \\ \Rightarrow L_{path}^{dB} &= L_{fs}^{dB} + L_{t,point}^{dB} + L_{pol}^{dB} + L_{atm}^{dB} \end{aligned} \quad (5.13)$$

Beim Empfänger werden ähnlich wie beim Sender der Antennengewinn  $G_r^{dB} = 16 \text{ dBi}$  sowie Leitungsverluste  $L_{r,line}^{dB} = -3 \text{ dB}$  berücksichtigt, um die empfangene Leistung zu bestimmen. Die Benennung der Variable für die empfangene Leistung erfolgt hier in

Übereinstimmung mit der vorherigen Darstellung als  $P_r$ . In der Literatur wird diese jedoch oft als  $C$  für Carrier, also die empfangene Leistung der Trägerwelle, bezeichnet (siehe Abschnitt 5.1). Um eine Verwechslung mit der Batteriekapazität auszuschließen, wird hier jedoch  $P_r$  verwendet. Darüber hinaus wird auch hier ein Pointing Loss der Empfangsantenne von  $L_{r,point}^{dB} = -2$  dB berücksichtigt. Die gesuchte empfangene Leistung ergibt sich unter Berücksichtigung der Verluste zu

$$P_r = EIRP \cdot G_r \cdot L_{path} \cdot L_{r,line} \cdot L_{r,point} \quad (5.14)$$

$$\Rightarrow P_r^{dBW} = EIRP^{dBW} + G_r^{dBi} + L_{path}^{dB} + L_{r,line}^{dB} + L_{r,point}^{dB} \quad (5.15)$$

### 5.3.3 Rauschen

Abschließend muss das Rauschen des Signals betrachtet werden. Jede Komponente eines Senders oder Empfängers erzeugt Rauschen aufgrund der thermischen Bewegung der Elektronen in der Komponente. Diese Bewegung bewirkt zufällige Spannungen oder Ströme, die eine gewisse Rauschleistung zur Folge haben. Die Rauschleistung  $N$  hängt daher von der Temperatur der Komponente  $T$  ab und ergibt sich näherungsweise zu

$$N = k \cdot T \cdot B \quad (5.16)$$

$$\Rightarrow N^{dB} = k^{dB} + T^{dB} + B^{dB} \quad (5.17)$$

wobei  $k = 1,381 \cdot 10^{-23} \frac{J}{K}$  die Boltzmann-Konstante und  $B$  die Rauschbandbreite ist. Die dabei verwendete Temperatur wird „effektive Rauschtemperatur“ genannt. Bei realen elektronischen Bauteilen, zum Beispiel im Empfänger eines Kommunikationssystems, kann diese Temperatur der realen Temperatur entsprechen. Einer allgemeineren Definition folgend, kann aber zu jedem Rauscheinfluss auf ein Kommunikationssystem eine effektive Rauschtemperatur angegeben werden, die nicht mit einer tatsächlich vorhandenen Temperatur übereinstimmen muss [19]. Somit können nicht nur durch Bauteile verursachte Rauschleistungen berücksichtigt werden, sondern auch solche, die auf dem Übertragungsweg des Signals auftreten, wie zum Beispiel durch Regen.

Die effektive Rauschtemperatur des Gesamtkommunikationssystems wird aufgeteilt in die sogenannte Antennenrauschteperatur und die Empfängerrauschteperatur [16]. Die Antennenrauschteperatur fasst dabei alle Rauscheinflüsse zusammen, die bis zu der Antenne des Empfängers auftreten. Dabei sind verschiedenste Quellen zu beachten. Eine der wichtigsten Quellen für Rauschen ist die Erde selber, aufgrund ihrer Temperatur von im Mittel 290 K. Außerdem sollte die Empfangsantenne nicht in die Nähe der Sonne ausgerichtet sein, da diese durch ihre sehr hohe Temperatur einen signifikanten Rauscheintrag verursacht. Auch bei einer Ausrichtung der Antenne auf den Mond kann eine Erhöhung der Rauschleistung gemessen werden. Ein gewisses Maß an kosmischem Hintergrundrauschen muss insbesondere bei niedrigen Frequenzen ebenfalls berücksichtigt werden. Weitere natürliche Rauschquellen sind Wolken, Regen und andere atmosphärische Einflüsse, insbesondere bei hohen Frequenzen. Dazu kommen noch Einflüsse durch andere Funk- oder Störsignale sowie gegebenenfalls durch Objekte auf der Funkstrecke. Die Antennenrauschteperatur für die



Bodenstation für CLAVIS wird im Linkbudget mit 405 K angegeben.

Die Empfängerrauschtemperatur beinhaltet alle Rauscheinflüsse im Empfänger, vor allem durch Leitungen und Verstärker. Kennt man das Verhältnis von Ein- und Ausgangsleistung sowie die ungefähre Temperatur der einzelnen Komponenten lassen sich die jeweiligen effektiven Rauschtemperaturen bestimmen. Im Linkbudget für die AISat-Konfiguration werden insbesondere die Einflüsse der Leitungen und des Verstärkers im Empfänger in der Bodenstation berücksichtigt. Diese verschiedenen Einflüsse können dann unter Berücksichtigung von Gewinn- und Verlustfaktoren zur effektiven Rauschtemperatur des Gesamtsystems  $T_{sys}$  aufsummiert werden [11]; für CLAVIS ergibt sich  $T_{sys} = 473$  K. Diese Temperatur kann dann in Gleichung (5.16) verwendet werden. Damit ergibt sich das gesuchte Verhältnis  $\frac{P_r}{N}$  unter Verwendung der Gleichungen (5.14) und (5.16) zu

$$\frac{P_r}{N} = \frac{EIRP \cdot G_r \cdot L_{path} \cdot L_{r,line} \cdot L_{r,point}}{k \cdot T_{sys} \cdot B} \quad (5.18)$$

wobei  $B$  hier die Bandbreite der Funkverbindung angibt. Diese entspricht ungefähr der doppelten Datenrate und wird mit  $B = 10$  kHz berücksichtigt. Ausgedrückt in Dezibel ergibt sich das CNR mit Hilfe der Gleichungen (5.15) und (5.17) zu

$$\left(\frac{P_r}{N}\right)^{dB} = EIRP^{dBW} + G_r^{dBi} + L_{path}^{dB} + L_{r,line}^{dB} + L_{r,point}^{dB} - k^{dB\frac{1}{K}} - T_{sys}^{dBK} - B^{dBHz} \quad (5.19)$$

Dabei wird oft das Verhältnis  $\frac{G_r}{T_{sys}}$  als Kennwert des Empfängers direkt angegeben, so zum Beispiel im Empfängermodell in STK (siehe Abschnitt C.2.7). Für CLAVIS ergibt sich mit den oben angegebenen Werten

$$\begin{aligned} \frac{G_r}{T_{sys}} &= 10^{\frac{16\text{dBi}}{10}} \cdot \frac{1}{473\text{K}} = 0,0842 \frac{1}{\text{K}} \\ \left(\frac{G_r}{T_{sys}}\right)^{dB\frac{1}{K}} &= 16\text{dBi} - (10 \cdot \log(473))\text{dBK} = -10,749\text{dB}\frac{1}{\text{K}} \end{aligned}$$

Das CNR kann dann nach Eingabe der Kennwerte von Sender und Empfänger sowie der Verluste für jeden Zeitpunkt einer Satellitenmission mit Hilfe von STK berechnet werden. Dies ist insbesondere sinnvoll, da die Entfernung zwischen Satellit und Bodenstation quadratisch in die Freiraumdämpfung eingeht und somit einen sehr großen Einfluss hat. Dieser kann in STK unter Berücksichtigung der genauen Form der Erde und der exakten Positionen von Satellit und Bodenstation besonders genau bestimmt werden. Damit das von CLAVIS ausgesendete frequenzmodulierte Signal fehlerfrei demoduliert werden kann, wird hier

$$\left(\frac{P_r}{N}\right)^{dB} > 10\text{dB}$$

gefordert.

# Kapitel 6

## Lageregelungssystem

### 6.1 Gesamtsystem

Wie eingangs schon erwähnt (siehe Kapitel 2), besitzt CLAVIS zur Lageregelung neben einem Ausleger zur Gravitationsgradientenstabilisierung drei Magnet Torquers, die ein Magnetfeld erzeugen, das mit dem Erdmagnetfeld interagiert, so dass eine zusätzliche Stabilisierung erreicht wird. Darüber hinaus kommen Magnetfeldsensoren zum Einsatz, die Stärke und Richtung des Erdmagnetfeldes bestimmen. Aus diesen Daten können dann eine geeignete Stärke und Richtung des durch die Magnet Torquer erzeugten Magnetfeldes bestimmt werden, so dass der Satellit in der gewünschten Lage gehalten wird.

Diese zusätzliche Stabilisierung ist zum einen direkt nach der Trennung des Satelliten von der Raketenoberstufe notwendig, da der Satellit zu diesem Zeitpunkt im allgemeinen Fall unkontrolliert um alle drei Achsen rotiert. Entfaltet man zu diesem Zeitpunkt die Helixantenne, würde sich zwar bei nicht allzu großen Rotationsgeschwindigkeiten eine grobe Ausrichtung in Nadirrichtung einstellen. Dabei existieren jedoch zwei gleichwahrscheinliche Fälle: Die Helixantenne kann in Richtung Nadir oder in Richtung Zenit zeigen. Um den letzteren Fall, bei dem keine Signale von der Erde empfangen werden könnten, zu vermeiden, werden die Magnet Torquer eingesetzt. Zum anderen müssen zu jedem Zeitpunkt auftretende Schwingungen um die nominelle Lage durch die Magnet Torquer ausgeglichen werden. Neben den schon erwähnten Rotationsbewegungen nach der Trennung von der Oberstufe können diese Schwingungen auch durch Unregelmäßigkeiten im Gravitationsfeld der Erde, solaren Strahlungsdruck, ungewollte Wechselwirkungen von elektronischen Komponenten mit dem Erdmagnetfeld und atomosphärischen Störungen hervorgerufen werden.

Im Rahmen dieser Arbeit ist die Lage des Satelliten vor allem für das Energieversorgungssystem von Bedeutung. Die zeitliche Änderung der Lage im Raum bewirkt eine Änderung des Einstrahlwinkels der Sonne auf die Solarpaneele und damit eine Änderung der bereitgestellten Leistung. Darüber hinaus kann die Abweichung von der Nadirausrichtung bei der AISat-Konfiguration im Moment des Austritts aus dem Erdschatten je nach Richtung eine verstärkte oder verringerte Abschattung der Solarpaneele hervorrufen, was sich ebenfalls auf die bereitgestellte Leistung auswirkt.

## 6.2 Gravitationsgradientenstabilisierung

Zum Verständnis der Gravitationsgradientenstabilisierung stelle man sich zwei Massen vor, die durch eine starre, masselose Struktur verbunden sind. Befindet sich dieses System im erdnahen Orbit in einer Lage, in der beide Massen den gleichen Abstand vom Gravitationszentrum haben, so ist es praktisch momentenfrei. Wird nun aber ein kleines Störmoment aufgebracht, zum Beispiel durch solaren Strahlungsdruck, so erhöht sich der Abstand der einen Masse zum Gravitationszentrum und der Abstand der anderen Masse verringert sich. Damit gerät das System aus dem Gleichgewicht, da auf die näher am Gravitationszentrum befindliche Masse eine größere Gravitationskraft wirkt. Darüber hinaus bleibt die orbitale Geschwindigkeit konstant, so dass sich aufgrund der Verringerung des Abstandes zum Gravitationszentrum auch die auf die Masse wirkende Fliehkraft verringert. Bei der Masse, deren Abstand vom Gravitationszentrum steigt, verhält es sich gerade umgekehrt: Die Fliehkraft nimmt zu und die Gravitationskraft ab. Beide Effekte bewirken ein zusätzliches Moment, dass die Achse des Systems in radialer Richtung in Bezug auf die Erde ausrichtet. Dabei sind stets beide Ausrichtungen energetisch gleich günstig, unabhängig von dem Betrag der Massen (siehe auch [20]).

Betrachtet man nun ein reales Raumfahrzeug, muss dessen Massenverteilung in Betracht gezogen werden. Dazu wird ein mitbewegtes Koordinatensystem definiert, dessen positive x-Achse in Flugrichtung und dessen positive z-Achse in Nadirrichtung zeigen; die y-Achse vervollständigt das Koordinatensystem senkrecht zur Orbitalenebene. Der Ursprung des Koordinatensystems liegt dabei im Massenmittelpunkt des betrachteten Raumfahrzeuges. Dieses Koordinatensystem wird als „Vehicle Velocity Local Horizontal“ (VVLH)-System bezeichnet. Das körperfeste Koordinatensystem des Raumfahrzeugs wird dazu so definiert, dass deren x-, y- und z-Achse bei einer idealen Nadirausrichtung gerade deckungsgleich mit denen des VVLH-Systems liegen.

Geht man nun davon aus, dass die Achsen des körperfesten Koordinatensystems den Hauptträgheitsachsen entsprechen, so dass der Trägheitstensor als Vektor  $(J_x, J_y, J_z)$  angegeben werden kann, so lässt sich die Gravitationsgradientenstabilisierung am besten ausnutzen, wenn  $J_z < J_x = J_y$  ist. Das bedeutet, dass das Massenträgheitsmoment um die z-Achse kleiner ist als die gleich großen Momente um x- und y-Achse. Bei homogener Massenverteilung würde dies einem länglichen zylindrischen Körper entsprechen, dessen Länge größer ist als der Durchmesser der Grundfläche. Liegt ein unsymmetrischer Körper mit  $J_x \neq J_y$  vor, können zusätzliche Störmomente auftreten. Auf realen Raumfahrzeugen wird die nötige Massenverteilung oft durch einen Ausleger oder eine längliche Form der Struktur erreicht. Auf der AISat-Konfiguration von CLAVIS dient dazu wie schon erwähnt die Helixantenne.

In der Praxis ist die Lage von Raumfahrzeugen, deren Lagestabilisierung ausschließlich auf dem Gravitationsgradienten beruht, niemals stabil. Jede Auslenkung aus der optimalen, in Richtung Nadir orientierten Lage führt zu einem Rückstellmoment aufgrund des Gravitationsgradienten, das wiederum eine Schwingung über die radiale Ausrichtung hinaus verursacht. Somit stellt sich eine Schwingungsbewegung um die

optimale Lage ein. Wie schon erwähnt, können verschiedene Einflüsse zu einer solchen Auslenkung führen, wobei insbesondere die rotationsellipsoide Gestalt der Erde eine bedeutende Rolle spielt. Diese bewirkt eine Schwingung um die  $y$ -Achse des VVLH-Systems, also innerhalb der Orbitalebene. Unter bestimmten Umständen, je nach Massenverteilung des Raumfahrzeugs und den vorhandenen Störmomenten, können sich die Schwingungen soweit verstärken, dass sie Amplituden von  $90^\circ$  erreichen und somit den Effekt der Gravitationsgradientenstabilisierung zunichte machen. Um dies zu verhindern, werden auf CLAVIS die Magnet Torquer eingesetzt.

### 6.3 Lageverhalten von CLAVIS

Zur Beschreibung der Lage eines Satelliten im Raum können in STK Eingabedateien verwendet werden, in denen die Winkel zwischen dem erdbezogenen VVLH-Koordinatensystem und dem körperfesten Koordinatensystem für diskrete Zeitpunkte angegeben werden, wobei zwischen den Zeitpunkten automatisch interpoliert wird (siehe auch Abschnitt 7.1.2). Dabei werden diese Winkel in dem mitbewegten, lokalen VVLH-System analog zu den Lagewinkeln eines Flugzeugs als Roll- (um die  $x$ -Achse), Nick- (um die  $y$ -Achse) und Gierwinkel (um die  $z$ -Achse) bezeichnet<sup>1</sup>. Datenreihen für Schwingungen um bestimmte Achsen lassen sich als einfache Sinusschwingungen angenähert mit Hilfe eines MATLAB-Skripts berechnen (siehe Abschnitt 8.2.2).

Detaillierte Simulationen des Lageverhaltens von CLAVIS in der AISat-Konfiguration unter Berücksichtigung des geplanten Orbits wurden an anderer Stelle durchgeführt [22]. Die Ergebnisse dieser Simulationen zeigen, dass die Magnet Torquer die Schwingungen um die gewünschte Lage weitgehend unterdrücken können. Es zeigt sich, dass eine Nickschwingung mit einer Amplitude von ca.  $5^\circ$  auftritt, die sich relativ genau zwei Mal pro Orbit wiederholt, was darauf zurückzuführen ist, dass sich die Veränderungen im Gravitationspotential aufgrund der rotationsellipsoiden Form der Erde ebenfalls zwei Mal pro Orbit wiederholen. Darüber hinaus zeigen die Simulationen eine ebenfalls recht langsame Rollschwingung mit einer Amplitude von ca.  $3^\circ$ . Außerdem rotiert der Satellit langsam um seine  $z$ -Achse.

---

<sup>1</sup>Die Bezeichnungen Roll-, Nick- und Gierwinkel werden auch verwendet, wenn die Lage eines Satelliten in einem inertialen Koordinatensystem, also zum Beispiel einem erdfesten System, angegeben wird. Dabei soll der Gegensatz zu den Eulerwinkeln betont werden, bei denen unterschiedliche Rotationen um die gegebenen Achsen verwendet werden. Diese Winkel sind jedoch strikt von den hier verwendeten zu trennen (siehe auch [21]).

# Kapitel 7

## Verwendung der Software

### 7.1 SATELLITE TOOL KIT

#### 7.1.1 Allgemeines

Zur Berechnung der gesuchten Daten, wie der Länge von Sonnenlichtabschnitten und Bodenkontaktzeiten, dem Einfallswinkel der Sonnenstrahlung auf die Solarpaneele etc., wird hier wie schon erwähnt die Software SATELLITE TOOL KIT (STK) der Firma ANALYTICAL GRAPHICS INC. (AGI) verwendet. Dabei handelt es sich um ein Softwaretool, mit dem die Bewegungen von allen möglichen Boden-, Luft- und Raumfahrzeugen simuliert und unter Berücksichtigung praktisch aller Umweltfaktoren visualisiert und analysiert werden können. Es besteht die Möglichkeit, beliebige interessierende Daten in sogenannten „Reports“ ausgeben zu lassen. Somit können nicht nur Positionen, Richtungen und daraus abgeleitete Daten wie zum Beispiel die durch Solarpaneele generierte Energie analysiert werden, sondern auch Informationen zu Beziehungen zwischen verschiedenen Objekten, wie die Sichtbarkeit von Satelliten von einer Bodenstation aus oder die Qualität einer bestehenden Funkverbindung, ausgewertet werden. Am Institut für Raumfahrtssysteme des DLR stand für diese Arbeit STK in der Version 9.2.1 zur Verfügung.

Um mit STK arbeiten zu können, muss zunächst ein Szenarioobjekt erstellt werden, dessen wichtigste Attribute der zu analysierende Zeitraum, ein Zeitintervall sowie der Zentralkörper - hier stets die Erde - sind. Darüber hinaus kann in den Eigenschaften des Szenarioobjekts eine Datei berücksichtigt werden, die die genaue Höhe der Erdoberfläche in Bezug zu dem verwendeten Referenzellipsoid in Abständen von 30 Winkelsekunden enthält. Als untergeordnete Objekte können nun alle möglichen Einrichtungen und Fahrzeuge erstellt werden. Im Rahmen dieser Arbeit werden Satelliten- (*Satellite*), Bodenstations- (*Facility*) sowie Sender- (*Transmitter*) und Empfängerobjekte (*Receiver*) verwendet. Die verschiedenen Objekttypen werden in den folgenden Abschnitten mit ihren wichtigsten Einstellungen vorgestellt. Weitere Informationen dazu finden sich in [21]. Da hier vor allem die bereitgestellten Daten und weniger die Darstellung der Missionsgeometrie im Vordergrund stehen, wird nicht im Detail auf die beträchtlichen Möglichkeiten der Visualisierung eingegangen. Darüber hinaus wird im Rahmen dieser Arbeit STK immer mit einem MATLAB-Skript (siehe Abschnitt 7.2) gesteuert, weshalb auch nicht auf die Details des GUI eingegangen wird.

### 7.1.2 Satellitenobjekt

Das Satellitenobjekt ist das für diese Arbeit wichtigste Objekt und außerdem das mit den meisten Manipulationsmöglichkeiten. Zunächst ist es wichtig, dem Objekt einen Orbit zuzuweisen. Dies kann über die Eingabe der KEPLERSchen Bahnelemente oder über einen Orbit-Wizard geschehen, in dem zum Beispiel SSOs durch die Eingabe der Bahnhöhe und der LTDN definiert werden können. Darüber hinaus wird auch dem Satelliten ein Zeitraum zugewiesen, während dem seine Bahn propagiert wird. Als Propagator steht neben einfachen analytischen  $J_2$ - bzw.  $J_4$ -Propagatoren auch der sogenannte „High Precision Orbit Propagator“ (HPOP) zur Verfügung, mit dem die Bahn unter Berücksichtigung praktisch aller relevanten Einflüsse bestimmt werden kann. Dazu werden jedoch genaue Angaben zur Massenverteilung und zur Gestalt des Satelliten, sowie ein genauer Startzeitpunkt zur Berücksichtigung des wechselnden Einflusses der Sonne auf die Atmosphäre sowie weitere Eingaben benötigt. Darüber hinaus ist diese Propagation im Vergleich zu den analytischen Verfahren deutlich rechenaufwändiger und damit langsamer. Bei der Bahnpropagation des Satelliten werden für den eingestellten Missionszeitraum die Positionen des Satelliten im Abstand des im Szenarioobjekt definierten Zeitintervalls in einer Datei abgelegt.

#### Definition der Lage

Die Lage des Satelliten im Raum kann auf zwei verschiedene Weisen definiert werden. Zum einen gibt es verschiedene voreingestellte Profile, von denen hier das Profil „Spin about Nadir“ verwendet wird, das den Satelliten in Nadirrichtung ausrichtet und eine Rotation um die radiale Achse zulässt. Die Geschwindigkeit dieser Rotation kann eingestellt werden, wobei auch die Möglichkeit besteht, diese ganz zu unterdrücken. Darüber hinaus kann ein Offset der Rotation angegeben werden. Belässt man diesen Wert bei null, entspricht das körperfeste Koordinatensystem des Satelliten dem VVLH-System; die körperfeste x-Achse zeigt in Flugrichtung, die körperfeste z-Achse zeigt Richtung Nadir (siehe auch Abschnitt 6.2). Gibt man einen Offset ungleich null an, sind die körperfesten x- und y-Achsen um den angegebenen Winkel gegen die x- und y-Achsen des VVLH-Systems in positiver mathematischer Drehrichtung verdreht, wobei die z-Achsen beider Systeme weiterhin die gleiche Richtung besitzen.

Neben diesen vorgegebenen Profilen lässt sich auch eine externe Datei (\*.a) erstellen, die Datenpunkte in bestimmten zeitlichen Abständen zur Lage des Satelliten enthält. Auch hier stehen eine Menge verschiedener Möglichkeiten zur Definition der Lage im Raum zur Verfügung. Die hier benutzte Variante wurde bereits in Abschnitt 6.3 dargestellt. Im Folgenden ist beispielhaft ein Auszug aus einer solchen \*.a-Datei gezeigt.

```
stk.v.5.0
```

```
BEGIN Attitude
```

```
ScenarioEpoch      15 Aug 2011 00:00:00.000000
NumberOfAttitudePoints 527041
BlockingFactor      20
InterpolationOrder  1
CentralBody         Earth
CoordinateAxes       Custom VVLH(Earth) Satellite/CLAVIS
Sequence            321
```

```
AttitudeTimeYPRAngles
```

```

0      0.000000    5.000000    0.000000
60     0.000000    4.958722    0.384692
120    0.000000    4.835569    0.763032
.
.
.
END Attitude

```

Darin wird zuerst die STK-Version genannt, für die die Datei erstellt wurde. Sie kann jedoch mit allen höheren Versionen verwendet werden. Daraufhin folgen die Schlüsselwörter *BEGIN Attitude*. Im Kopf der Datei wird dann die Epoche als Datum angegeben. Darauf folgt die Anzahl der im Hauptteil enthaltenen Datenpunkte. Der „Blocking Factor“ gibt den für das Auslesen der Datei reservierten Speicher an. Daraufhin wird die Ordnung des Interpolationsverfahrens definiert, die hier um aufwändige Rechnungen zu vermeiden auf eins gesetzt wird, so dass eine lineare Interpolation verwendet wird. Als Zentralkörper wird wiederum die Erde verwendet. Die angegebenen Koordinatensysteme sind das auf die Erde bezogene VVLH-System und das körperfeste Koordinatensystem von CLAVIS. Der folgende Punkt gibt die Reihenfolge der Drehwinkel an, in diesem Fall wird zuerst der Gier-, dann der Nick- und als letztes der Rollwinkel angegeben. Vor dem Hauptteil der Datei wird die Art der angegebenen Winkel definiert. Der hier dargestellte Kopf der Datei wird im Rahmen dieser Arbeit stets so belassen, lediglich die Epoche kann sich ändern. Der Hauptteil der Datei besteht aus Datenpunkten in vier Spalten, wobei die erste Spalte die Zeit nach der Epoche in Sekunden und die anderen die vorher spezifizierten Roll-, Nick- und Gierwinkel (in dieser Reihenfolge) angeben. Schließlich folgen die Schlüsselwörter *END Attitude*.

### 3D-Modell

Die Lage des Satelliten ist hier vor allem zur Bestimmung der Winkel zwischen Sonneneinstrahlung und den Solarpaneelen von Bedeutung. In STK kann dem Satelliten ein dreidimensionales Modell zugewiesen werden, das ebenfalls über eine externe Datei definiert wird (\*.mdl). Beispieldateien für die Standard- und die AISat-Konfiguration finden sich im Anhang D.1. Darin können primitive Körper wie Zylinder, Quader etc. in ihrer Gestalt und Lage über Eingabe von Koordinaten definiert werden. Somit lässt sich die Grundstruktur eines Satelliten aufbauen, wobei beachtet werden muss, dass bei der Nadirausrichtung immer die positive z-Achse des Satelliten in Richtung Nadir zeigt, was wiederum für die Eingabe der Koordinaten der einzelnen Komponenten von Bedeutung ist. In der dabei verwendeten Skriptsprache können einzelne Komponenten als Solarpaneele gekennzeichnet werden, so dass die erzeugte Energie berechnet werden kann. Die Erstellung einer solchen Datei ist in der STK Hilfe [21] beschrieben.

### Solar Panel Tool

Zur Bestimmung der von der Sonne angestrahlten Paneelflächen steht in STK das sogenannte „Solar Panel Tool“ zur Verfügung, das ein grafisches Verfahren zur Bestimmung der für die Energieerzeugung relevanten Daten verwendet. Über einen gegebenen Zeitraum wird dabei der Satellit von der Sonne aus betrachtet in einem eigenen Fenster vor einem roten Hintergrund dargestellt. Alle Teile des Satelliten,

die keine Solarpaneele sind, erscheinen schwarz. Solarpaneele erscheinen je nach Winkel zur Sonneneinstrahlung in verschiedenen Abstufungen zwischen rot wie der Hintergrund (Sonneneinstrahlung senkrecht) und schwarz (Sonneneinstrahlung parallel). Anders gesagt zeigt die Farbe eines Paneels die effektive Fläche an, die zur Energieerzeugung genutzt werden kann, also die Fläche multipliziert mit dem Kosinus des Einfallswinkels (siehe Abschnitt 4.3.1).

Das Tool zählt bei der Ausführung zu jedem Zeitpunkt in Abständen des angegebenen Zeitintervalls die sichtbaren Pixel aller Solarpaneele, gegebenenfalls unter Berücksichtigung einer möglichen Abschattung der Paneele durch Antennen oder andere Komponenten des Satelliten. Da die Gesamtfläche bekannt ist, kann daraus die gesamte angestrahlte Fläche und die effektiv angestrahlte Fläche jedes Paneels ausgegeben werden. Dabei kann ein „Bound Radius“ eingestellt werden, der die Größe des im Analysefenster dargestellten Ausschnitts angibt. Dieser sollte so gewählt werden, dass der Satellit einen möglichst großen Teil des Fensters einnimmt, da die Genauigkeit der Ausgaben mit zunehmender Pixelzahl steigt. Der Satellit sollte aber nicht über die Begrenzung des Fensters hinausragen, da die Zählung der Pixel ansonsten fehlerhaft wird. Für CLAVIS hat sich ein Wert von 0,5m für den Bound Radius bei maximiertem Fenster als sinnvoll herausgestellt. Darüber hinaus sollte die Option „Offscreen Rendering“ gewählt werden, da ansonsten nur Pixel gezählt werden, die für den Benutzer sichtbar sind. Befindet sich ein anderes Fenster über dem Analysefenster oder wird STK minimiert, verfälscht sich die Analyse ansonsten extrem.

Grundsätzlich muss beachtet werden, dass es sich hierbei um ein numerisches Verfahren handelt, das niemals ein exaktes Ergebnis liefern kann. Insbesondere bei niedrigen Einfallswinkeln kann die Pixelzahl eines Paneels sehr klein werden, so dass sich die Genauigkeit verringert. Das Solar Panel Tool ist dennoch sehr nützlich, um den Einfluss der Abschattung durch Komponenten des Satelliten abschätzen zu können. Da es sich um ein grafisches Verfahren handelt, ist es allerdings auch verhältnismäßig rechenaufwändig. Ist keine Abschattung durch Antennen oder ähnliches zu erwarten, wie zum Beispiel bei der Standardkonfiguration, wird hier deshalb auf ein analytisches Verfahren zurückgegriffen (siehe Abschnitt 8.3.2).

### 7.1.3 Bodenstationsobjekt

Ein Bodenstationsobjekt wird über seine geografische Lage definiert. Es besteht die Möglichkeit aus einem Katalog eine Stadt auszuwählen, deren Koordinaten voreingestellt sind. Darüber hinaus kann die Lage aber auch in verschiedenen Koordinatensystemen manuell eingegeben werden. Hier wird die Eingabe in geodätischen Koordinaten bevorzugt, wobei anstatt die exakte Höhe einzugeben, die Option „Use Terrain“ gewählt wird. Damit wird die Höhe der hinzugefügten Bodenstation so definiert, dass sie der Höhe der Erdoberfläche über dem Referenzellipsoid an der gegebenen geografischen Position entspricht. Die dazu notwendigen Daten finden sich in der entsprechenden Datei, die im Szenarioobjekt ausgewählt werden kann (siehe Abschnitt 7.1.1).



Das Bodenstationsobjekt kann dazu benutzt werden, die Kontaktzeiten zu einem Satelliten sowie Azimut, Elevation und Abstand des Satelliten von der Bodenstation aus gesehen auszugeben. Es können jedoch keine Details einer Sende- oder Empfangsanlage definiert werden, was auch bei einem Satellitenobjekt nicht möglich ist. Dazu werden die im folgenden Abschnitt beschriebenen Sender- und Empfängerobjekte verwendet.

#### 7.1.4 Sender- und Empfängerobjekt

Die Sender- und Empfängerobjekte können Satelliten oder Bodenstationen als untergeordnete Objekte zugeordnet werden und befinden sich dadurch automatisch immer am selben Ort wie das darüberliegende Objekt. Es stehen verschiedene Modelle zur Verfügung in denen die Parameter für den Sender oder den Empfänger gesetzt werden können. Dabei wird hier nur das jeweils einfachste Modell verwendet („Simple Receiver/Transmitter Model“), was ausreichend ist, um die in Abschnitt 5.3 angegebenen Kennwerte einzugeben. Damit kann das gesuchte CNR bestimmt und ausgegeben werden.

In den Eigenschaften des Senderobjekts können die Sendefrequenz, die EIRP und die Bandbreite des Übertragungskanal eingestellt werden. Beim Empfängerobjekt kann das Verhältnis von Antennengewinn zur Systemrauschtemperatur direkt eingegeben werden. Die Sendefrequenz und die Bandbreite werden automatisch auf die Werte des Senderobjekts gesetzt, sofern sie nicht explizit angegeben werden. Bei beiden Objekten besteht die Möglichkeit, weitere konstante Verluste in einer Liste manuell einzufügen, die dann bei der Berechnung des CNR zusätzlich zur Freiraumdämpfung berücksichtigt werden. All diese Werte können für die AISat-Mission dem Linkbudget entnommen werden [18].

## 7.2 Schnittstelle STK-MATLAB

In MATLAB können mit Hilfe einer einfachen Skriptsprache Berechnungen auf der Basis von Vektoren und Matrizen durchgeführt werden. Darüber hinaus besteht die Möglichkeit, mit MATLAB Eingaben in STK zu tätigen, also zum Beispiel Szenarios aufzubauen, und von STK ausgegebene Daten zu speichern und weiterzuverarbeiten. Am Institut für Raumfahrtssysteme des DLR stand für diese Arbeit MATLAB in der Version 7.9.0 (R2009b) zur Verfügung. Die Skripte werden als \*.m-Datei abgespeichert und können entweder als direkt ausführbare Programme oder als Funktionen, die von anderen Skripten aufgerufen werden können, implementiert sein. Die grundlegende Funktionalität zum Erstellen von STK-Szenarios und Abrufen von Daten aus STK mit MATLAB ist im Anhang C beschrieben.

# Kapitel 8

## Durchführung der Simulationen

### 8.1 Übersicht

Die im Rahmen dieser Arbeit durchgeführten Simulationen lassen sich in zwei Gruppen einteilen. Zunächst wird die AISat-Mission detailliert betrachtet, da für diese Mission die wichtigsten Eckdaten bekannt sind. Im Anschluss werden verschiedenste mögliche CLAVIS-Missionen weniger detailliert untersucht, um grobe Anhaltswerte für zukünftige Missionsauslegungen zu liefern. Die folgenden Abschnitte enthalten die Eingabedaten dieser Simulationen und die getroffenen Annahmen.

#### 8.1.1 AISat-Mission

Die AISat-Mission soll wie schon in Abschnitt 2.3 erwähnt, im August 2011 gestartet werden. Da der genaue Starttermin während der Durchführung dieser Arbeit noch nicht feststand, wird hier der 15. August 2011, 0h UTC als Start der Mission angenommen. Der Orbit ist ein SSO mit einer LTDN von 10h in einer Höhe von 650 km, wobei hier wie auch in der gesamten Arbeit von einem kreisförmigen Orbit ausgegangen wird. Mit Hilfe des Orbit Wizards von STK wurde daraus die RAAN zum Startzeitpunkt zu  $293,06^\circ$  bestimmt. Das Argument des Perigäums sowie die wahre Anomalie werden zu Beginn der Mission auf  $0^\circ$  belassen. Die Mission wird über eine Dauer eines Jahres betrachtet. Der Orbit wird sowohl mit einem einfachen  $J_2$ -Propagator als auch mit dem HPOP (siehe Abschnitt 7.1.2) propagiert und die Ergebnisse werden gegenübergestellt.

Für die AISat-Mission wird ein Modell der entsprechenden Konfiguration von CLAVIS geladen, so dass die von jedem Paneel bereitgestellte Leistung mit Hilfe des Solar Panel Tools (siehe Abschnitt 8.3.1) über die gesamte Missionsdauer bestimmt werden kann. Dabei werden vier verschiedene Szenarien des Lageverhaltens betrachtet:

- exakte Nadirausrichtung der Helixantenne
- Schwingungen um  $\pm 5^\circ$  in Nick- und  $\pm 3^\circ$  in Rollrichtung
- Schwingungen um  $\pm 15^\circ$  in Nick- und  $\pm 9^\circ$  in Rollrichtung
- exakte Zenitausrichtung der Helixantenne

Die exakte Nadirausrichtung stellt dabei den theoretisch optimalen Fall dar, die Schwingungen um  $5^\circ$  und  $3^\circ$  den realistischen Fall (siehe Abschnitt 6.3). Die Schwingungen größerer Amplitude repräsentieren ein ursprünglich angenommenes Worst-Case-Szenario. In beiden Szenarien wird angenommen, dass es sich bei den Schwingungen um harmonische Schwingungen handelt, die sich zwei Mal pro Orbit wiederholen. Wie in Abschnitt 6.2 beschrieben, richtet sich der Satellit nach der Trennung von der Oberstufe mit gleicher Wahrscheinlichkeit mit der Helixantenne in Richtung Nadir oder in Richtung Zenit aus. Letzteres soll natürlich durch die Magnet Torquer verhindert werden, was aber zu Beginn der Mission nicht garantiert werden kann. Daher wird hier auch dieser Fall betrachtet, um die dann zur Verfügung stehende Energie abschätzen zu können. Die Rotation des Satelliten wird wie in Abschnitt 8.2.1 beschrieben berücksichtigt. Die Parameter des Energieversorgungssystems, insbesondere der Solarpaneele, wurden in Kapitel 4 definiert. Zur Bestimmung der von der Sonne gelieferten Leistung pro Fläche im Erdorbit wird Gleichung (4.2) herangezogen.

Zur Betrachtung der Datenübertragung zur Bodenstation wird für AISat nur die Bodenstation in Bremen berücksichtigt. Dabei werden alle Bodenkontaktzeiten unter Berücksichtigung der minimalen Elevation berechnet sowie das CNR des Downlinks zu verschiedenen Zeitpunkten während der Bodenkontaktzeiten. Daraus kann dann die Kontaktzeit berechnet werden, während der das CNR groß genug ist, damit das Signal fehlerfrei empfangen werden kann. Die verwendeten Parameter der Funkverbindung sind in Abschnitt 5.3.2 dargestellt. Eine detaillierte Erläuterung der MATLAB-Skripte zur Simulation der AISat-Mission findet sich im Anhang E. Darin ist auch beispielhaft der Quellcode eines Skripts wiedergegeben.

### 8.1.2 Zukünftige CLAVIS-Missionen

Für zukünftige Missionen steht weder Orbit noch Startzeitpunkt fest. Wie in Kapitel 3 gezeigt wird, hängt der Verlauf der Eklipsenzeiten und damit prinzipiell auch der Verlauf der erzeugten Energie auf kreisförmigen Orbits von der Bahnhöhe, der Inklination, der RAAN sowie vom Startzeitpunkt ab. Beim Startzeitpunkt spielt jedoch das Jahr nur eine Rolle für den Einfluss der Sonne auf die Atmosphäre. Wichtiger ist die Unterscheidung der Jahreszeit bzw. des Tages im Jahr, an dem der Start stattfindet. Alle diese Parameter des Orbits werden in den Simulationen variiert.

Die Simulationen werden mit Hilfe von zwei verschiedenen Skripten durchgeführt, wobei eines die Orbits mit Inklinationen unter  $90^\circ$  abdeckt und ein anderes die SSOs. Der Zeitraum des STK-Szenarios wird vom 01.01.2012 bis zum 01.01.2014 gewählt. Der Startzeitpunkt der einzelnen simulierten Missionen wird dann innerhalb des Jahres 2012 variiert, so dass die Missionen mit einer Dauer von einem Jahr innerhalb des Zeitraums des Szenarios enden. Im einzelnen werden die Bahnhöhe zwischen 200 km und 650 km in 50 km-Schritten, die Inklination zwischen  $0^\circ$  und  $90^\circ$  in  $5^\circ$ -Schritten, die RAAN zwischen  $0^\circ$  und  $360^\circ$  in  $20^\circ$ -Schritten und der Startzeitpunkt zwischen dem 01.01.2012 und dem 31.12.2012 in Schritten von 21 Tagen variiert. Damit ergeben sich insgesamt 64980 betrachtete Fälle.

Bei den SSOs werden RAAN, Startzeitpunkt und Bahnhöhe auf die gleiche Weise variiert, wobei auch die entsprechende LTDN bestimmt wird. Die Inklination hingegen wird Gleichung (3.10) folgend als Funktion der jeweiligen Bahnhöhe berechnet. Damit ergeben sich 3420 verschiedene Fälle.

Zur Propagation der Orbits wird jeweils ein  $J_2$ -Propagator verwendet, da aufgrund der Vielzahl der betrachteten Fälle die Rechenzeit eine wichtige Rolle spielt. Darüber hinaus sind Parameter wie die Masse des Satelliten und die Aktivität der Sonne zum Startzeitpunkt nicht verfügbar, so dass der Einfluss der Atmosphäre für zukünftige Missionen nur sehr grob abgeschätzt werden kann. Die erhaltenen Werte für niedrige Orbits, auf denen der atmosphärische Widerstand einen signifikanten Einfluss hat, können dann als Richtwerte für die entsprechende Bahnhöhe genommen werden. Wird CLAVIS für eine zukünftige Mission auf eine entsprechend niedrige Umlaufbahn gebracht, können der Verlauf der Bahnhöhe über der Zeit im Voraus bestimmt und die hier bereitgestellten Werte als Anhaltspunkte für die Energieerzeugung und die Bodenkontaktzeiten verwendet werden. Die Ergebnisse einer einfachen Simulation der Evolution der Bahnhöhe für verschiedene Fälle finden sich als Referenz in Abschnitt 10.7.

Aufgrund der Vielzahl der betrachteten Fälle ist es nicht möglich, die Verläufe der erzeugten Energie oder anderer Parameter über ein ganzes Jahr darzustellen. Daher werden hier für jeden Fall jeweils der Worst Case sowie ein durchschnittlicher Wert der erzeugten Energie als Anhaltspunkte bestimmt. Darüber hinaus werden die Dauer der kürzesten, mittleren und längsten Sonnenlichtphase berechnet sowie die gesamte Bodenkontaktzeit zu den Bodenstationen in Bremen und Inuvik. Da Einzelheiten des Kommunikationssystems ebenfalls nicht bekannt sind, wird hier das Linkbudget nicht weiter betrachtet.

In den Skripten wird ein analytisches Verfahren zur Berechnung der erzeugten Energie der Solarpaneele verwendet, was wiederum die Rechenzeit verringert (siehe Abschnitt 8.3.2). Es kann hier eingesetzt werden, da nach derzeitigem Stand keine Antennen oder andere Ausleger die Solarpaneele der hier betrachteten Konfigurationen abschatten. Dabei werden die Standardkonfiguration sowie die Konfiguration mit ausklappbaren Paneelen betrachtet, wobei ein Ausklappwinkel der Paneele von  $60^\circ$  angenommen wird. Hierbei werden die Werte für das obere Paneel und die seitlichen Paneele stets getrennt berechnet. Damit können ebenfalls die Werte für die Konfiguration mit um  $90^\circ$  ausgeklappten Paneelen bestimmt werden, indem die durch das obere Paneel erzeugte Energie als Ausgangswert verwendet wird. Auch Konfigurationen mit fehlenden Paneelen können somit betrachtet werden (siehe auch Abschnitt 8.2.1). Da die betrachteten schlechtesten und durchschnittlichen Fälle für die Energieerzeugung zu jeder Jahreszeit auftreten können, wird hier der geringste Wert der von der Sonne eingestrahlt Leistung pro Fläche angenommen. Dieser tritt auf, wenn sich die Erde im Aphel ihrer Bahn befindet, also Anfang Juli, und ergibt sich nach Gleichung (4.2) zu  $1325,8 \frac{\text{W}}{\text{m}^2}$ .

Bei diesen Simulationen wird stets eine exakte Nadirausrichtung des Satelliten angenommen, da auch Details zur Lageregelung nicht bekannt sind. Darüber hinaus

werden aber Werte zur Abschätzung der Reduktion der erzeugten Energie durch Schwingungsbewegungen angegeben. Das zur Simulation der zukünftigen CLAVIS-Missionen verwendete MATLAB-Skript findet sich im Anhang F.

Mit Hilfe der Ergebnisse einer weiteren Simulation wird außerdem der Zusammenhang zwischen der Dauer der Sonnenlichtperiode und der erzeugten Energie für die verschiedenen Konfigurationen betrachtet. Dazu ist ein zusätzliches Simulationsskript nötig, in dem nur die große Halbachse und die Inklination variiert werden, da somit der gesamte Bereich an möglichen Sonnenlichtperiodendauern abgedeckt wird. Dabei wird die Inklination in  $15^\circ$ -Schritten und die große Halbachse in 50 km-Schritten in den oben erwähnten Bereichen variiert und jeweils Missionen über ein Jahr bei konstantem Startzeitpunkt und -rektaszension simuliert. Zu allen Sonnenlichtperioden wird dabei die Dauer und die erzeugte Energie abgespeichert, so dass der Zusammenhang analysiert werden kann. Das entsprechende MATLAB-Simulationsskript findet sich ebenfalls im Anhang F.

## 8.2 Getroffene Annahmen

Zur Vereinfachung der Bestimmung der erzeugten Energie werden hier verschiedene Annahmen bezüglich der Rotation und der Roll- und Nickschwingungen des Satelliten getroffen. Diese werden in den folgenden Abschnitten beschrieben.

### 8.2.1 Rotation

Wie in Abschnitt 6.3 beschrieben, wird die AISat-Konfiguration relativ langsam um die in Nadirrichtung ausgerichtete Achse rotieren. Ein ähnliches Verhalten ist auch bei den anderen Konfigurationen je nach verwendetem Lageregelungssystem wahrscheinlich. Grundsätzlich ist diese Rotation problemlos in den Simulationen zu berücksichtigen, jedoch kann die Drehrate für die zukünftigen Missionen nicht abgeschätzt werden. Bei Testläufen der Simulationen ergab sich darüber hinaus ein weiteres Problem. Betrachtet man zum Beispiel den Verlauf der erzeugten Energie eines Paneels über ein Jahr ergibt sich unter Vernachlässigung der Rotation ein stetiger Verlauf. Berücksichtigt man eine Rotation ist dieser Verlauf stark „verrauscht“, so dass wesentliche Merkmale kaum noch erkennbar sind. Durch diesen mehr oder weniger zufälligen Einfluss kann sich auch die Berechnung von Durchschnittswerten bzw. die Angabe von Minimal- und Maximalwerten verfälschen. Es ist praktisch nicht möglich, die Rotation für die hier durchgeführten Simulationen so vorzugeben, dass die resultierenden Berechnungen charakteristische, aussagekräftige Werte ergeben, wie zum Beispiel einen Worst Case oder einen durchschnittlichen Fall. Daher wird hier ein anderer Ansatz für die Berücksichtigung der Rotation verwendet.

Zum Verständnis dieses Ansatzes stelle man sich CLAVIS in der Standardkonfiguration in einem Orbit vor, dessen Betawinkel gleich  $90^\circ$  ist, so dass die Sonne senkrecht auf die Orbitalebene scheint. Dabei rotiert der Satellit nicht, ist exakt in Richtung Nadir ausgerichtet und ein Solarpaneel zeigt genau in Flugrichtung, so dass praktisch nur ein Paneel von der Sonne mit einem Einfallswinkel von  $90^\circ$  beschienen wird. Auf

diesem Orbit stellt dies den schlechtest möglichen Fall dar, da jedes weitere, schräg beschienene Paneel die bereitgestellte Leistung erhöht. Ist der Satellit nun um  $45^\circ$  gedreht, ist der optimale Fall erreicht, da zwei Paneele beschienen werden mit einem Einfallswinkel von jeweils  $45^\circ$ . Ist der Satellit um weitere  $45^\circ$  gedreht, also insgesamt um  $90^\circ$ , ist wieder der schlechteste Fall nur eines beschienenen Paneels erreicht. Hier wird nun davon ausgegangen, dass die erzeugte Leistung der seitlichen Paneele als Funktion des Drehwinkels im allgemeinen Fall einer Kosinusfunktion entspricht.

Kosinus- und Sinusfunktionen haben die Eigenschaft, dass der Mittelwert aus zwei Funktionswerten, die sich im Abstand einer halben Periode der Funktion befinden, dem Mittelwert der gesamten Funktion entspricht. Diese Eigenschaft wird hier ausgenutzt. Dazu wird die durch die Solarpaneele erzeugte Energie für den Fall bestimmt, dass der Rotationswinkel über die gesamte Mission konstant gleich  $0^\circ$  ist, also das körperfeste Koordinatensystem des Satelliten dem VVLH-System entspricht (siehe Abschnitt 7.1.2). Die gleiche Berechnung wird durchgeführt für den Fall, dass der Rotationswinkel, also der Offset des Lageprofils Spin about Nadir in STK,  $45^\circ$  entspricht. Schließlich wird der Mittelwert aus beiden Werten bestimmt. Dieser entspricht dann unter Voraussetzung der genannten Annahmen dem Mittelwert der von einem rotierenden Satelliten erzeugten Energie.

Diese Betrachtungsweise hat einen weiteren Vorteil. Setzt man eine Rotationsbewegung voraus, so sind die seitlichen Paneele praktisch gleichberechtigt und müssen nicht unterschieden werden, da jedes Paneel über einen längeren Zeitraum betrachtet gleich lange in der Sonne und im Schatten ist. Da die Gesamtenergieproduktion der seitlichen Paneele bekannt ist, kann somit auch die von einer Konfiguration mit einem fehlenden Paneel erzeugte Energie überschlägig bestimmt werden, indem ein Viertel der von den seitlichen Paneelen Energie abgezogen wird.

### 8.2.2 Roll- und Nickschwingungen

Wie in Abschnitt 8.1 dargelegt, werden die Roll- und Nickschwingungen nur bei der Simulation der AISat-Mission direkt berücksichtigt. Hier ist der mögliche negative Einfluss auch deutlich größer als bei den anderen Konfigurationen, da die Reflektorplatte beim Austritt aus dem Erdschatten eine zusätzliche Abschattung der Paneele bewirken kann, wenn die Ausrichtung des Satelliten gerade ungünstig ist.

Die Frequenz der Nickschwingung des Satelliten beträgt wie in Abschnitt 6.3 beschrieben recht genau zwei pro Orbit. Hier wird davon ausgegangen, dass die Rollschwingung die gleiche Frequenz besitzt, was auch ungefähr dem in der Simulation des Lageverhaltens erhaltenen Wert entspricht. Darüber hinaus wird hier die Annahme getroffen, dass es sich bei diesen Schwingungen um harmonische Schwingungen handelt, so dass sie durch einfache Sinusfunktionen beschrieben werden können. Diese Annahmen haben wie später gezeigt wird zur Folge, dass sich die erzeugte Energie des schwingenden Satelliten im Vergleich zu einem exakt in Richtung Nadir ausgerichteten periodisch verändern, so dass regelmäßig Minima und Maxima durchlaufen werden (siehe Abschnitt 9.2.2). Damit kann der Einfluss der Schwingungen auf einem bekannten Orbit relativ gut angegeben werden. Wären die

Schwingungen nicht in Resonanz mit der Umlaufzeit, wäre dies nicht möglich.

Zur Erzeugung der entsprechenden Attitude-Dateien (\*.a) für STK wurde die MATLAB-Funktion *attitude\_file\_generator.m* erstellt. Diese Funktion erzeugt eine STK-konforme Attitude-Datei (siehe Abschnitt 7.1.2) unter Angabe der gewünschten Epoche, der Amplituden der Roll- und Nickschwingungen, dem gewünschten Ablagepfad der \*.a-Datei, der großen Halbachse des Orbits sowie der gewünschten Zeitspanne, für die die Date gültig sein soll. Darin kann außerdem der Rotationswinkel des Satelliten auf einen konstanten Wert gesetzt werden, so dass die zwei Fälle  $0^\circ$  und  $45^\circ$  (siehe vorheriger Abschnitt) berücksichtigt werden können. Die Erzeugung der Datenreihen für die Schwingungen erfolgt alle 60 Sekunden unter Verwendung einfacher Sinus- und Kosinusfunktionen. Damit beschreibt die Helixantenne des Satelliten in der Simulation eine Ellipse um die Nadirrichtung, die sich zwei Mal pro Orbit wiederholt. Der Quellcode der Funktion findet sich im Anhang D.2.

### 8.2.3 Verwendung des HPOP

Als Vergleich zu dem bei den Simulationen meistens verwendeten  $J_2$ -Propagator wird zur Simulation der AISat-Mission und zur Abschätzung der Verringerung der Bahnhöhe mit der Zeit auch der HPOP von STK herangezogen. Dieser Propagator berücksichtigt neben den Störungen durch das Erdpotential auch die Abbremsung durch die Atmosphäre, Drittkörperstörungen, Störungen durch Strahlungsdruck etc. Um damit verlässliche Werte zu erhalten, sind einige besondere Einstellungen notwendig, die in diesem Abschnitt beschrieben werden. Alle nicht erwähnten Größen und Einstellungen werden auf den Standardeinstellungen von STK belassen. Die für diese Einstellungen nötigen Befehlszeilen finden sich im Anhang C.4. Genauere Ausführungen zu den Bahnstörungen finden sich zum Beispiel in [6].

Zunächst werden Start- und Endzeitpunkt, Zeitschritt sowie der zu propagierende Orbit analog zum  $J_2$ -Propagator definiert (siehe Abschnitt C.2.4). Standardmäßig werden die Störungen durch das unregelmäßige Erdgravitationspotential mit harmonischen Koeffizienten bis zum 21. Grad und zur 21. Ordnung berücksichtigt (siehe auch [4]); diese Einstellung wird beibehalten. Außerdem werden Bahnabweichungen aufgrund von Gezeitenkräften auf den festen Erdkörper berücksichtigt.

Für die Betrachtung des Einflusses der Atmosphäre wird der Widerstandsbeiwert des Satelliten auf dem Standardwert 2,2 belassen. Zur Bestimmung des Flächen- zu Massenverhältnisses wird CLAVIS als Würfel mit der Kantenlänge 30 cm angenähert. Als mittlere angeströmte Fläche wird das 1,5-fache einer Seitenfläche angenommen, also  $1,5 \cdot (30 \text{ cm})^2 = 0,135 \text{ m}^2$ . Das Gewicht von CLAVIS wird mit 13 kg angenommen. Damit ergibt sich ein Flächen- zu Massenverhältnis von ca.  $0,0104 \frac{\text{m}^2}{\text{kg}}$ . Zur Modellierung des Einflusses der Sonne und des Erdmagnetfeldes auf die Atmosphäre wird eine von STK bereitgestellte Datei verwendet, die entsprechende gemessene und vorausgesagte Werte enthält (*stkFluxGeoMag.fxm*). Das voreingestellte Atmosphärenmodell Jacchia-Roberts [21] wird beibehalten. Da dieses aber nur für Höhen über 90 km definiert ist, kann nur bis zu dieser Höhe propagiert werden. Dies ist für

Rechnungen für niedrige Startorbits von Bedeutung; die AISat-Mission erreicht solch geringe Höhen nicht innerhalb ihrer nominellen Missionsdauer (siehe auch Abschnitt 10.7).

Um den Einfluss des solaren Strahlungsdrucks berücksichtigen zu können, wird zunächst ebenfalls ein Flächen- zu Massenverhältnis benötigt. Die angestrahlte Fläche ist dabei im allgemeinen nicht identisch mit der durch die Restatmosphäre angeströmten Fläche. Für die angeströmte Fläche wurde hier jedoch eine durchschnittliche Querschnittsfläche des Würfels angenommen, weshalb der oben genannte Wert auch für den solaren Strahlungsdruck verwendet wird. Außerdem muss ein Koeffizient angegeben werden, der die Reflexionseigenschaften des Satelliten beschreibt. Dieser liegt zwischen null und zwei, wobei ein Wert von null einer vollständigen Transmission der Strahlung und ein Wert von zwei einer vollständigen Reflexion der Strahlung entspricht. Dieser Wert ist im allgemeinen nicht konstant und umständlich zu bestimmen. In [27] wird jedoch ein grober Richtwert von 1,21 für Solarpaneele, die den Großteil der Außenfläche von CLAVIS ausmachen, angegeben. Dieser Wert wird hier verwendet. Außerdem wird die Option *Use Boundary Mitigation* verwendet, mit der eine Fehlerkorrektur beim Übergang vom Erdschatten in die Sonne vorgenommen wird [21].

Darüber hinaus wird auch die von der Erde ausgehende Strahlung in die Berechnungen der Störungen durch den Strahlungsdruck einbezogen. Die Erde strahlt zum einen auf der Tagseite das Sonnenlicht zu einem gewissen Maße zurück (Albedo) und strahlt Wärmestrahlung im Infrarotbereich von ihrer gesamten Oberfläche ab. Das dabei verwendete Flächen- zu Massenverhältnis wird mit der Fläche der Reflektorplatte von ca.  $0,2734 \text{ m}^2$  zu  $0,0210 \frac{\text{m}^2}{\text{kg}}$  bestimmt, da die Reflektorplatte bei Nadirausrichtung stets Richtung Erde zeigt. Der Reflexionskoeffizient wird auf den gleichen Wert wie beim solaren Strahlungsdruck gesetzt.

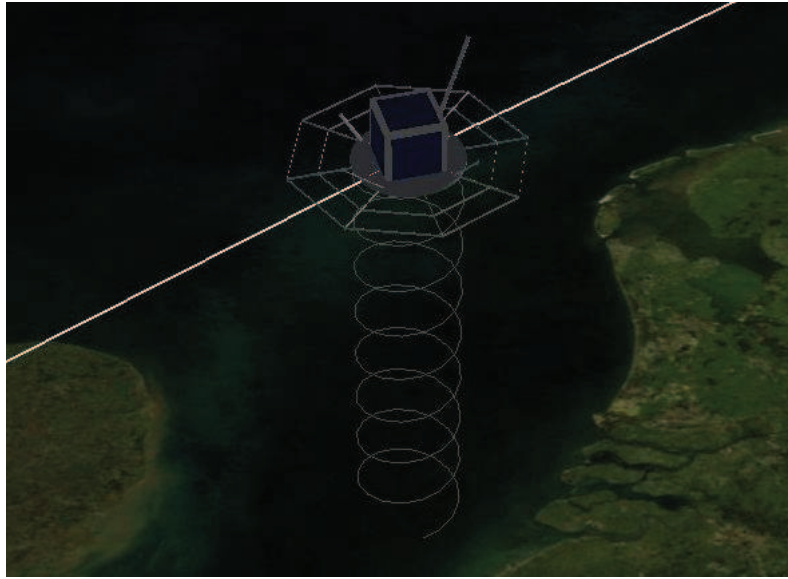
## 8.3 Berechnung der erzeugten Energie

Zur Berechnung der während einer Sonnenlichtperiode von den einzelnen Paneelen erzeugten Energie wird hier in Abständen von einer Minute die erzeugte Leistung der Paneele bestimmt und aufintegriert. Dazu werden wie bereits erwähnt zwei verschiedene Verfahren eingesetzt: ein numerisches für die AISat-Mission (Solar Panel Tool) und ein analytisches für die zukünftigen Missionen anderer Konfigurationen. Die beiden Verfahren werden im Folgenden genauer beschrieben.

### 8.3.1 Energieberechnungen über das Solar Panel Tool

Damit das Solar Panel Tool verwendet werden kann, wurde ein Modell des Satelliten als \*.mdl-Datei erstellt (siehe Abschnitt 7.1.2). Die entsprechende Datei findet sich im Anhang D.1. Die Gestalt des Satelliten in dem Modell wurde weitestgehend dem CAD-Modell der AISat-Konfiguration nachempfunden, wobei die Geometrien vereinfacht dargestellt werden. Abbildung 8.1 zeigt die grafische Darstellung des Modells im STK GUI. Darin erkennt man, dass in diesem Modell zwei kurze Peitschenantennen auf





**Abbildung 8.1:** Grafische Darstellung des Modells der AISat-Konfiguration in STK

der Reflektorplatte angebracht sind, obwohl es in der Realität nur eine Antenne ist (siehe Abschnitt 2.3). Damit aber durch die fehlende Antenne die wahre Abschattung der Paneele nicht unterschätzt wird, wurde hier ein symmetrisches Modell erstellt. Dabei sind die blauen Flächen die ausgewiesenen Solarpaneele, die im Modell jeweils eine Fläche von  $A_{pan} = (26 \cdot 26) \text{ cm}^2 = 676 \text{ cm}^2$  besitzen. Da die erzeugte Energie später über die Fläche der Solarpaneele berechnet wird, muss dabei beachtet werden, dass die Fläche der Solarzellen pro Paneel nur  $A_{sol} = 14 \cdot 26,62 \text{ cm}^2 = 372,68 \text{ cm}^2$  beträgt (siehe Abschnitt 4.1), sofern die Standardsolarpaneele verwendet werden. Hier wird angenommen, dass die Solarzellen symmetrisch und gleichförmig auf den Paneelen angeordnet sind, so dass das Verhältnis der Solarzellenfläche zu der Paneelfläche als konstanter Faktor in die Berechnung einfließt. Diese Annahme gilt relativ gut, wie in Abbildung 4.1 zu erkennen ist. Der zu berücksichtigende Faktor  $F_{area}$  ergibt sich für die Standardsolarpaneele zu

$$F_{area} = \frac{A_{sol}}{A_{pan}} = \frac{372,68 \text{ cm}^2}{676 \text{ cm}^2} = 0,5513 \quad (8.1)$$

Als Ausgaben des Solar Panel Tools werden hier der Zeitpunkt, der Anteil der sichtbaren Sonnenscheibe sowie für jedes Paneel die angestrahlte Fläche und die effektive angestrahlte Fläche ausgegeben. Letztere entspricht dabei der angestrahlten Fläche multipliziert mit dem Kosinus des Einstrahlwinkels [21] (siehe auch Abschnitt 4.3.1). Da hier jedoch ein komplizierterer Zusammenhang zwischen Einstrahlwinkel und erzeugter Energie angenommen wird, ist der Einstrahlwinkel selbst von Interesse. Dieser kann aber nur unter Angabe eines diskreten Zeitpunktes aus STK ausgegeben werden, was eine sehr aufwändige Definition der Ausgaben nach sich ziehen würde. Daher wird der Einstrahlwinkel  $\gamma$  für ein Paneel aus der angestrahlten Fläche  $A_{ill}$  und der effektiv angestrahlten Fläche  $A_{eff}$  mit folgender Gleichung bestimmt.

$$\gamma = \arccos \left( \frac{A_{eff}}{A_{ill}} \right) \quad (8.2)$$

Wie in Abschnitt 7.1.2 beschrieben, kann die vom Solar Panel Tool ausgegebene Fläche insbesondere bei großen Einstrahlwinkeln ungenau sein. Um eine Überschätzung der erzeugten Energie zu vermeiden wird daher stets überprüft, ob die angegebene angestrahlte Fläche eines Paneels größer ist als  $A_{pan}$ . Ist dies der Fall, wird diese Fläche gleich  $A_{pan}$  gesetzt, bevor der Einstrahlwinkel berechnet wird. Mit dem berechneten Einstrahlwinkel kann dann der zu berücksichtigende Verlustfaktor nach Gleichung (4.3) berechnet werden.

Die zu jedem Zeitpunkt von den einzelnen Paneelen generierte Leistung kann unter Annahme eines konstanten Solarzellenwirkungsgrades  $\eta_{ges}$  (siehe Abschnitt 4.3.4) über Gleichung (4.4) bestimmt werden, wobei nun  $A_{ill}$  der aus dem Solar Panel Tool ausgegebenen beleuchteten Fläche entspricht und daher  $F_{area}$  als zusätzlicher Faktor berücksichtigt werden muss. Die über eine Sonnenlichtperiode erzeugte Leistung kann dann numerisch integriert werden, indem die Leistung zu jedem Zeitpunkt mit dem Zeitintervall multipliziert und die erhaltenen Werte aufaddiert werden. Die integrierte Größe besitzt dann die Einheit einer Energie.

### 8.3.2 Analytische Berechnungen der erzeugten Energie

Die analytische Berechnung der erzeugten Energie kann bei den anderen Konfigurationen eingesetzt werden, da keine Antennen oder ähnliches eine Abschattung der Paneele verursachen können. Dabei macht man sich zunutze, dass der Vektor  $\vec{s}$ , der von dem Satelliten zur Sonne zeigt, im körperfesten Koordinatensystem des Satelliten in STK berechnet werden kann. Kennt man darüber hinaus den Normalenvektor des betrachteten Solarpaneels  $\vec{n}$ , kann der Winkel zwischen diesen Vektoren  $\phi$  über die folgende bekannte Beziehung berechnet werden zu

$$\phi = \arccos \left( \frac{\vec{s} \bullet \vec{n}}{|\vec{s}| \cdot |\vec{n}|} \right) \quad (8.3)$$

Dieser Winkel entspricht im Grunde genommen dem Einfallswinkel  $\gamma$ , kann jedoch auch Werte annehmen, die größer sind als  $90^\circ$ . Ist dies der Fall, wird der entsprechend berechnete Winkel auf  $90^\circ$  gesetzt, so dass die erzeugte Leistung gleich null ist.

Die Normalenvektoren der Paneele können für die Standardkonfiguration und die Konfiguration mit ausgeklappten Paneelen im  $60^\circ$ -Winkel direkt angegeben werden. Bei der Standardkonfiguration entsprechen die Normalenvektoren der seitlichen Paneele den Einheitsvektoren in positiver bzw. negativer x- und y-Richtung. Der Normalenvektor des oberen Paneels entspricht der negativen z-Richtung. Die Vektoren lauten daher im körperfesten Koordinatensystem des Satelliten:

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}; \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}; \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

Für die Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen sind die Normalenvektoren der seitlichen Paneele im  $60^\circ$ -Winkel gegen die x- und y-Koordinatenachse geneigt. Der Normalenvektor des oberen Paneels bleibt gleich. Die Normaleneinheitsvektoren

der Solarpaneele dieser Konfiguration lauten im körperfesten Koordinatensystem des Satelliten:

$$\begin{pmatrix} 0,5 \\ 0 \\ -\frac{\sqrt{3}}{2} \end{pmatrix}; \begin{pmatrix} -0,5 \\ 0 \\ -\frac{\sqrt{3}}{2} \end{pmatrix}; \begin{pmatrix} 0 \\ 0,5 \\ -\frac{\sqrt{3}}{2} \end{pmatrix}; \begin{pmatrix} 0 \\ -0,5 \\ -\frac{\sqrt{3}}{2} \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

Ein weiterer Vorteil dieser Berechnungsmethode liegt darin, dass eine Rotation des Satelliten durch eine Veränderung der Normalenvektoren der Solarpaneele dargestellt werden kann. Definiert man diese Vektoren so, dass die Oberflächen der Paneele damit im  $45^\circ$ -Winkel zu den Koordinatenachsen liegen, hat man die Drehung um  $45^\circ$  erreicht, die zur Berücksichtigung der Rotation des Satelliten verwendet wird (siehe Abschnitt 8.2.1). Die so verdrehten Normaleneinheitsvektoren der Solarpaneele der Standardkonfiguration lauten damit:

$$\begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{pmatrix}; \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \\ 0 \end{pmatrix}; \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{pmatrix}; \begin{pmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \\ 0 \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

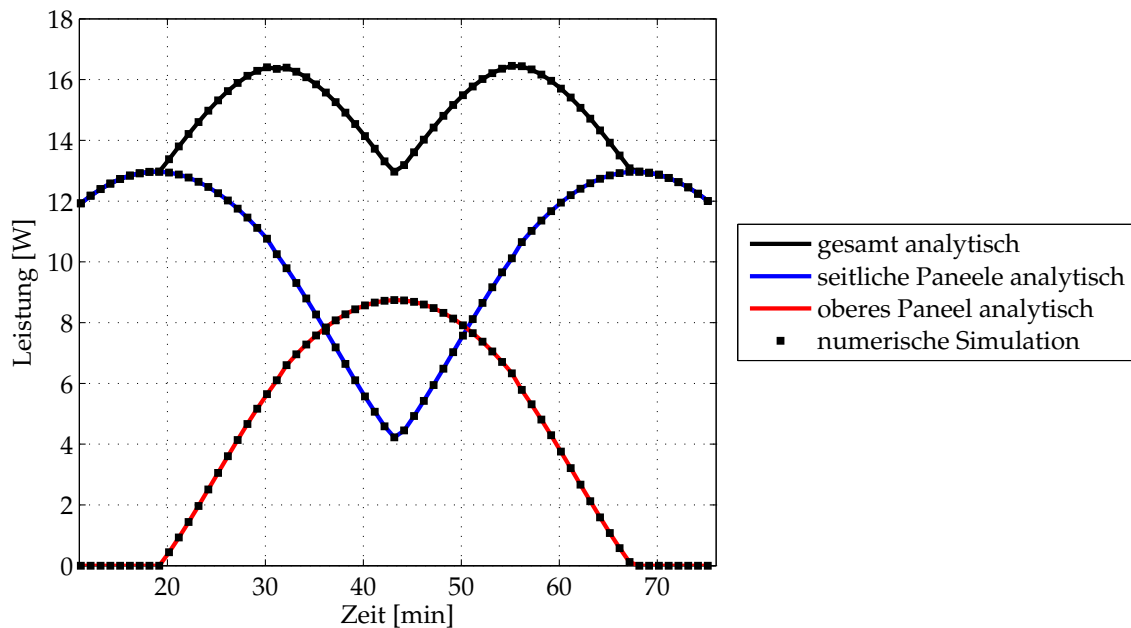
Die Normaleneinheitsvektoren der um  $45^\circ$  verdrehten Solarpaneele der Konfiguration mit ausklappbaren Paneelen lauten im körperfesten Koordinatensystem des Satelliten:

$$\begin{pmatrix} \frac{\sqrt{2}}{4} \\ \frac{\sqrt{2}}{4} \\ -\frac{\sqrt{3}}{2} \end{pmatrix}; \begin{pmatrix} -\frac{\sqrt{2}}{4} \\ -\frac{\sqrt{2}}{4} \\ -\frac{\sqrt{3}}{2} \end{pmatrix}; \begin{pmatrix} -\frac{\sqrt{2}}{4} \\ \frac{\sqrt{2}}{4} \\ -\frac{\sqrt{3}}{2} \end{pmatrix}; \begin{pmatrix} \frac{\sqrt{2}}{4} \\ -\frac{\sqrt{2}}{4} \\ -\frac{\sqrt{3}}{2} \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

Unter Verwendung dieser Vektoren und dem Einheitsvektor in Richtung Sonne lässt sich der Einfallswinkel der Sonne auf die Paneele und daraus der zu berücksichtigende Verlustfaktor  $F_{incid}$  nach Gleichung (4.3) berechnen. Damit kann Gleichung (4.4) angewendet werden, um die von einem Paneel zur Verfügung gestellte Leistung zu berechnen, wobei hier für die beleuchtete Fläche  $A_{ill}$  die Gesamtfläche der Solarzellen eingesetzt wird. Die durch alle vier seitlichen Paneele einer Konfiguration bereitgestellte Leistung wird für beide Drehwinkel ( $0^\circ$  und  $45^\circ$ ) aufsummiert und das Ergebnis halbiert. Damit wird die durchschnittliche von allen seitlichen Paneelen bereitgestellte Leistung bestimmt. Analog zu der numerischen Berechnung der Energie kann auch hier die während einer Sonnenlichtphase erzeugte Energie aus den zu jedem Zeitpunkt bestimmten Leistungen berechnet werden (siehe Abschnitt 8.3.1).

### 8.3.3 Vergleich der beiden Methoden

Um die beiden Berechnungsverfahren vergleichen zu können, wurde ein einfaches 3D-Modell der Standardkonfiguration erstellt. Die entsprechende \*.mdl-Datei findet sich im Anhang D.1. Die durch die einzelnen Paneele bereitgestellte Leistung wurde dann über eine Sonnenlichtperiode auf einem SSO, der dem Orbit der AISat-Mission gleicht, mit Hilfe des Solar Panel Tools bestimmt. Dabei wurde stets davon ausgegangen, dass der Aufbau aller Solarpaneele dem der Standardkonfiguration entspricht, um die Werte vergleichbar zu belassen. Darüber hinaus wurde die Leistung analytisch berechnet.



**Abbildung 8.2:** Vergleich der Berechnung der bereitgestellten Leistung der Solarpaneele der Standardkonfiguration über eine Sonnenlichtphase in einem SSO in 650 km Höhe mit einer LTDN von 10 h

Dabei wurde nur der Fall eines Rotationswinkels von  $0^\circ$  bei exakter Nadirausrichtung betrachtet. Die Ergebnisse sind in Abbildung 8.2 wiedergegeben. Darin ist jeweils der Verlauf der Leistung nach der analytischen Berechnung als Punkte und der Verlauf der numerischen Berechnung folgend als durchgezogene Linie dargestellt.

Man erkennt eine sehr gute Übereinstimmung der beiden Berechnungsmethoden auf allen Paneelen, woraus geschlossen werden kann, dass beide Verfahren richtige Ergebnisse liefern. Das zur Erstellung des Diagramms verwendete MATLAB-Skript findet sich im Anhang D.3. Die einzelnen Verläufe sind in Abschnitt 9.1 näher erläutert.

# Kapitel 9

## Ergebnisse für die AISat-Mission

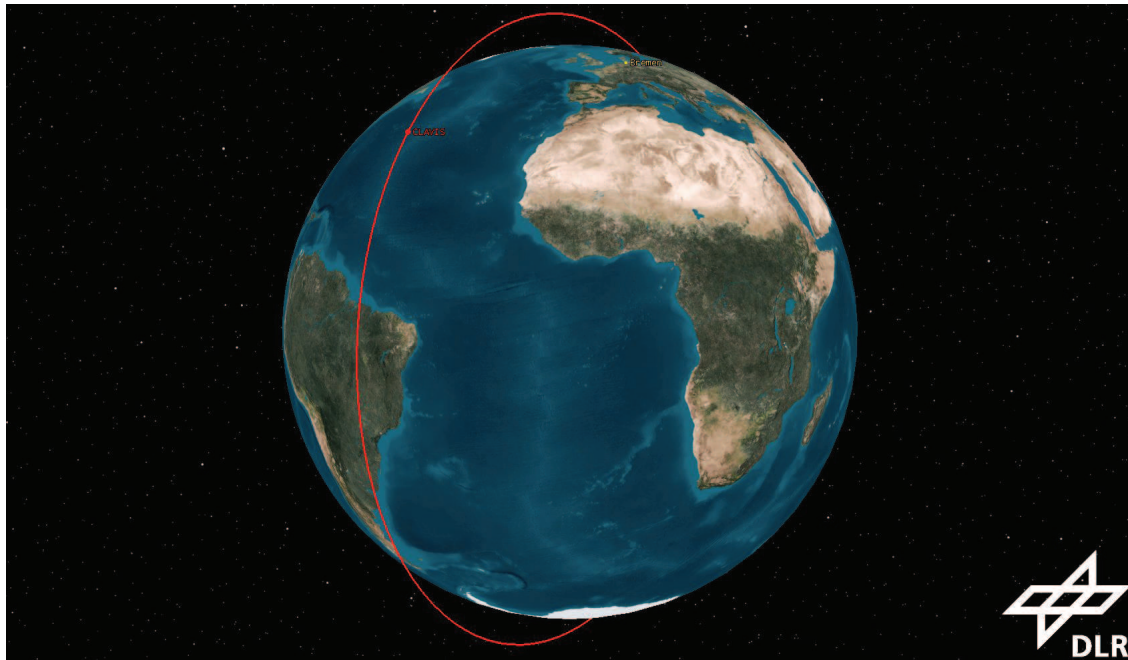
In den folgenden Abschnitten werden die Ergebnisse der Simulation der AISat-Mission vorgestellt und diskutiert (siehe Abschnitt 8.1.1). Dabei wird zunächst auf Verläufe über einen Orbit eingegangen, wobei auch ein Vergleich zwischen den verschiedenen Konfigurationen angestellt wird. Daraufhin wird die gesamte Mission betrachtet. Die dabei verwendeten MATLAB-Skripte werden im Anhang E näher erläutert und wiedergegeben.

### 9.1 Verläufe über einen Orbit

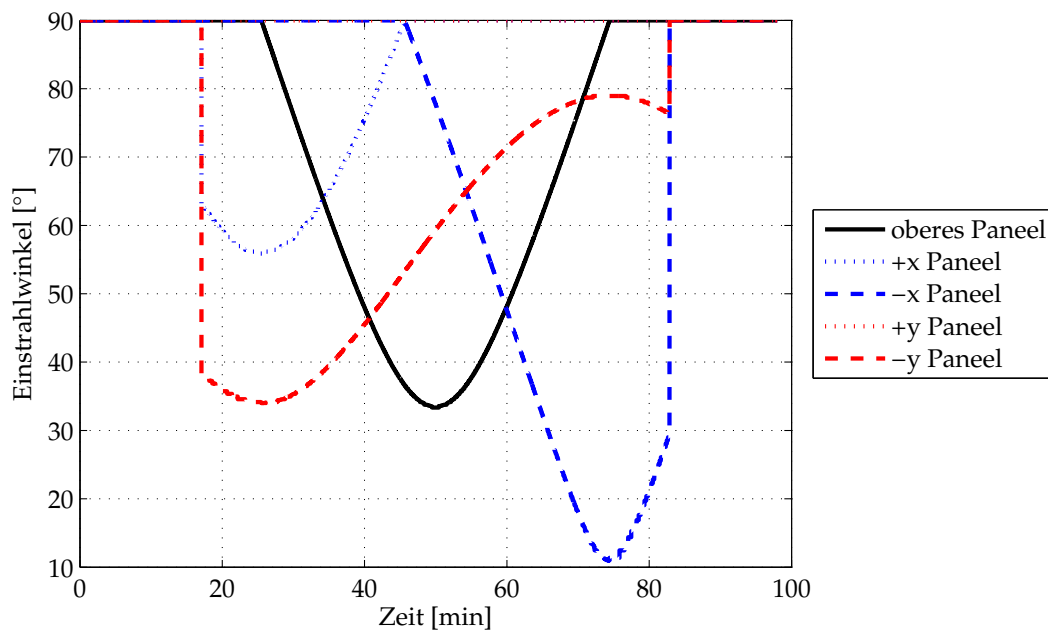
Wie in Abschnitt 3.3.4 dargelegt, sind die Schwankungen der Eklipsendauer auf SSOs mit einer LTDN im Bereich von 12 h sehr gering. Auf dem 10 h-Orbit der AISat-Mission sind die Schwankungen wie in Abschnitt 9.2.1 gezeigt wird ebenfalls relativ gering. Daher wird hier beispielhaft ein Orbit im März 2012 betrachtet, um Verläufe über einen Orbit darzustellen. Der betrachtete Orbit beginnt am aufsteigenden Knoten im Erdschatten und endet dort wieder. Die Umlaufdauer beträgt ca. 98 Minuten. Der Zeitpunkt im Jahr ist so gewählt, dass die Eklipsendauer ungefähr einem durchschnittlichen Wert im Laufe der einjährigen AISat-Mission entspricht. Der Orbit ist in Abbildung 9.1 aus der Richtung der Sonneneinstrahlung betrachtet dargestellt, wobei sich der absteigende Knoten im Vordergrund befindet. Die Bewegungsrichtung des Satelliten im Bild ist also von oben nach unten. Bei der Simulation wurde ein fester Drehwinkel von  $22,5^\circ$  bei exakter Ausrichtung in Richtung Nadir angenommen, um aussagekräftige Richtwerte zu erhalten (siehe auch Abschnitt 8.2.1). Die pro Fläche von der Sonne eingestrahelte Leistung wurde dabei auf den geringsten Wert von  $1325,8 \frac{\text{W}}{\text{m}^2}$  gesetzt. Als Zeitschritt wurde hier eine Sekunde gewählt, um einen genauen Verlauf zu erhalten.

#### 9.1.1 Verlauf der Einstrahlwinkel

Abbildung 9.2 zeigt zunächst den Verlauf des Einstrahlwinkels auf die verschiedenen Paneele. Dieser wird auf  $90^\circ$  gesetzt, wenn das entsprechende Paneel nicht beschienen wird und ist gleich  $0^\circ$ , wenn die Sonne senkrecht auf das Paneel scheint (siehe auch Abschnitt 4.3.1). Zu Beginn der Sonnenphase sind zunächst das Paneel in positiver x-Richtung (+x) und ein anderes seitliches Paneel (-y) in der Sonne. Der Einstrahlwinkel



**Abbildung 9.1:** SSO der AISat-Mission Anfang März aus Richtung der Sonne betrachtet

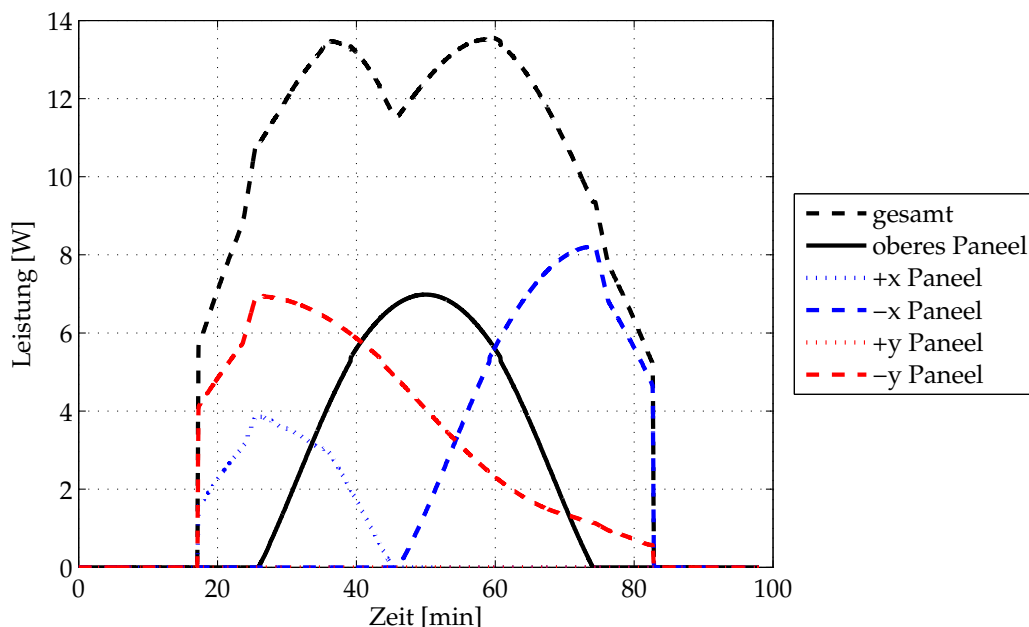


**Abbildung 9.2:** Verläufe des Einstrahlwinkels auf die verschiedenen Solarpaneele der AISat-Konfiguration von CLAVIS über einen Orbit

auf diese Paneele erreicht wenig später sein Minimum und steigt danach wieder an. Da der Satellit nicht rotiert, bleibt das -y-Paneele stets in der Sonne und das gegenüberliegende (+y) ist nie in der Sonne, da bei dieser Konfiguration zwei gegenüberliegende Paneele niemals gleichzeitig beschienen werden können. Der Einstrahlwinkel auf das Paneele in positiver x-Richtung steigt an, während sich der Satellit dem absteigenden

Knoten nähert. Da das Paneel um  $22,5^\circ$  von der Sonne „weggedreht“ ist, erreicht der Einstrahlwinkel die  $90^\circ$  noch bevor der Knoten erreicht ist. Zu diesem Zeitpunkt beginnt dann folglich der Einstrahlwinkel auf das gegenüberliegende -x-Paneel zu sinken, der in der Folge den günstigsten Einstrahlwinkel erreicht. Da dieses Paneel dabei stärker beschienen wird, verringert sich gleichzeitig der Einstrahlwinkel auf das -y-Paneel. Da die Sonne zu diesem Zeitpunkt im Jahr ungefähr senkrecht auf den Äquator scheint, ist der Verlauf des Einstrahlwinkels auf das obere Paneel näherungsweise symmetrisch und erreicht sein Minimum am absteigenden Knoten der Bahn.

### 9.1.2 Verlauf der erzeugten Leistung

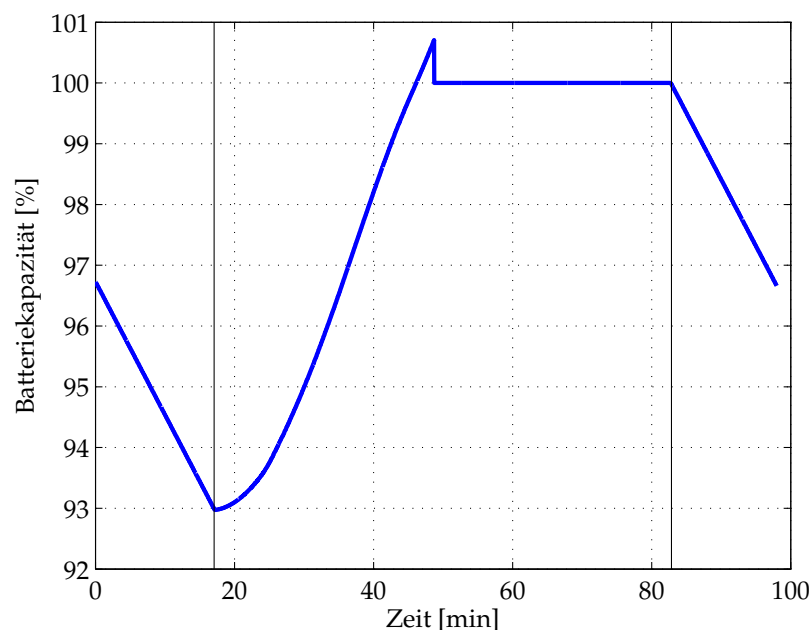


**Abbildung 9.3:** Verläufe der erzeugten Leistung der verschiedenen Solarpaneele der AISat-Konfiguration von CLAVIS über einen Orbit

Abbildung 9.3 zeigt die aus den Einstrahlwinkeln und den bestrahlten Flächen abgeleiteten Verläufe der von den einzelnen Solarpaneelen erzeugten Leistung sowie die aufsummierte Gesamtleistung. Die Verläufe für die einzelnen Paneele sind grundsätzlich denen der Einstrahlwinkel entgegengesetzt; je kleiner der Einfallswinkel, umso größer die erzeugte Leistung. Zu Beginn und gegen Ende der Sonnenlichtphase sind die Verläufe jedoch durch abrupte Änderungen der Kurvensteigung gekennzeichnet. Dies ist auf die während des Ein- und Austritts in das Sonnenlicht verstärkte Abschattung der Paneele durch die Reflektorplatte und die Radialantenne zurückzuführen (siehe Abschnitt 2.3). Sprünge in den Verläufen während der Sonnenlichtphase werden durch die Peitschenantennen verursacht, die ebenfalls die Paneele abschatten. In der Umgebung des absteigenden Knotens gibt es keine Abschattungen, da hier die Sonne von oben auf den Satelliten scheint, so dass sich keine Instrumente im Strahlengang befinden.

Aus der Addition der erzeugten Leistungen der einzelnen Paneele ergibt sich die Gesamtleistung. Deren Verlauf zeigt zwei Maxima vor und nach dem Erreichen des absteigenden Knotens, wenn das obere Paneel und die seitlichen Paneele zusammen die maximale Leistung erreichen.

### 9.1.3 Verlauf der Ladung der Batterie



**Abbildung 9.4:** Verlauf der Ladung der Batterie der AISat-Konfiguration von CLAVIS über einen Orbit bei konstanter Leistungsaufnahme, die senkrechten Linien deuten den Beginn und das Ende der Sonnenlichtperiode an

Abbildung 9.4 zeigt den Verlauf des Ladestandes der Batterie über einen Orbit unter Berücksichtigung der auftretenden Verluste (siehe Abschnitt 4.4). Die senkrechten Linien kennzeichnen den Anfang und das Ende der Sonnenlichtperiode. Dabei wurde von einer konstanten Leistungsaufnahme des Gesamtsystems von 4,29 W ausgegangen (siehe Abschnitt 4.2), weshalb der Verlauf der Batteriekapazität während der Zeit im Schatten linear ist. Bei der Berechnung des Startwertes wurde davon ausgegangen, dass die vorherige Schattenphase gerade so lange dauerte, wie die beiden Schattenphasen während des betrachteten Orbits zusammen. Die Batteriekapazität und -spannung werden auf den in Abschnitt 4.4 angegebenen konstanten Werten belassen. Der hier gezeigte Verlauf ist aufgrund der ungleichmäßigen Leistungsaufnahme so nicht realistisch. Hier soll vor allem der qualitative Verlauf dargestellt werden, wobei der Maximalwert der DoD einen vernünftigen Anhaltspunkt liefert.

Bei Eintritt in das Sonnenlicht wird zu jedem Zeitpunkt auf Grundlage der zur Verfügung stehenden Energie die resultierende Auf- oder Entladung der Batterie berechnet. In den ersten Sekunden der Zeit in der Sonne reicht die durch die Solarpaneele erzeugte Leistung nicht aus, alle Instrumente mit Strom zu versorgen, weshalb sich



die Batterie weiter entlädt. Der niedrigste Stand ist ca. sieben Sekunden nach Eintritt in das Sonnenlicht erreicht und entspricht einer DoD von ca. 7 %, was deutlich unter dem geforderten Maximalwert von 20 % liegt (siehe Abschnitt 4.1). Daraufhin wird die Batterie aufgeladen, was zunächst relativ langsam vonstattengeht, da die Paneele noch abgeschattet werden. Im weiteren Verlauf nimmt die Steigung der gezeigten Kurve zu und erreicht ihren Maximalwert am ersten Maximum der gesamten zur Verfügung stehenden Leistung, das auf Abbildung 9.3 zu erkennen ist. An diesem Punkt beträgt die momentane Aufladerate unter den hier getroffenen Annahmen ca.  $0,25 \cdot C_{batt}^*$ , was ebenfalls deutlich unter dem Maximum von  $1 \cdot C_{batt}^*$  liegt (siehe Abschnitt 4.4).

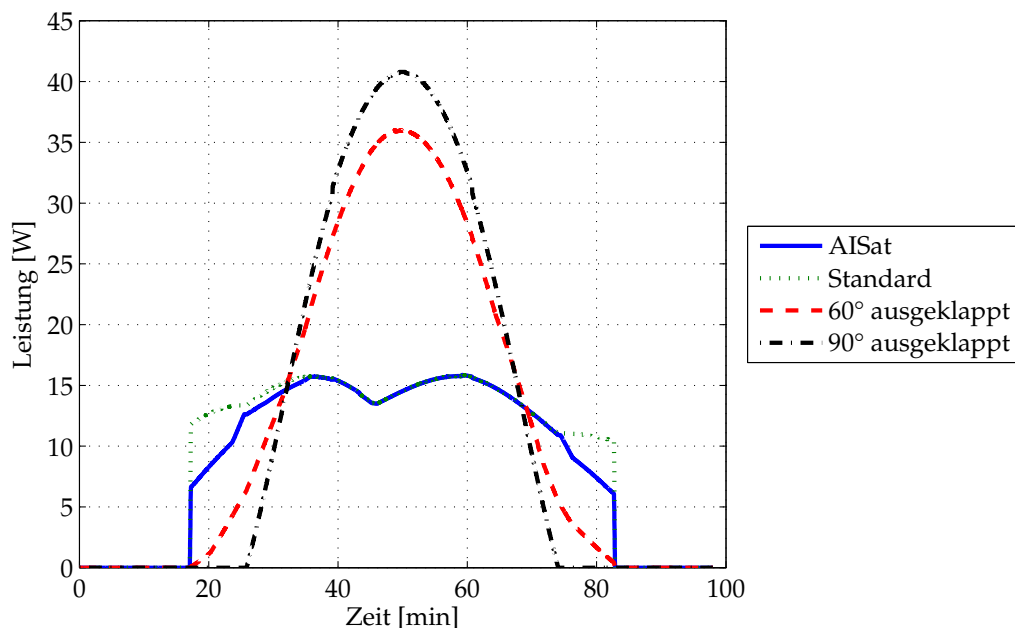
Daraufhin wird die Batterie weiter geladen, bis der rechnerische Ladestand etwas über 100 % beträgt. Dabei wird der Battery Return Factor von 1,1 berücksichtigt, was bedeutet, dass das 1,1-fache der entnommenen Kapazität wieder aufgeladen wird, um eine volle Ladung der Batterie sicherzustellen (siehe Abschnitt 4.1). Ist dieser Wert erreicht, wird hier die Kapazität der Batterie auf 100 % gesetzt und für den weiteren Verlauf während der Sonnenlichtzeit belassen. Die Aufladung der Batterie ist wie aus der Abbildung hervorgeht ungefähr nach der Hälfte der Zeit im Sonnenlicht abgeschlossen. In dem darauf folgenden Zeitabschnitt wird die in den Paneelen generierte Energie als Wärme dissipiert. In den letzten Sekunden der Sonnenlichtphase reicht die zur Verfügung stehende Energie wiederum nicht mehr aus, um das Gesamtsystem zu versorgen, weshalb der Ladestand der Batterie wieder zu sinken beginnt.

Der hier gezeigte Verlauf des Ladestandes der Batterie gilt streng genommen nur für den betrachteten Zeitpunkt, unterliegt jedoch auch weiteren Annahmen zu den Kenngrößen der Batterie und der Solarzellen sowie der Leistungsaufnahme der Instrumente. Daher sind die genannten Werte nur als Anhaltspunkte zu verstehen. Da alle kritischen Werte jedoch deutlich unter den Schwellenwerten liegen und sich die Dauer der Eklipsen über das Jahr kaum ändert, kann die Aussage getroffen werden, dass zu jedem Zeitpunkt der Mission genug Energie für den nominellen Betrieb zur Verfügung steht. Dies wird auch durch die weiteren Ergebnisse bestätigt, die in den folgenden Abschnitten vorgestellt werden.

#### 9.1.4 Vergleich der verschiedenen Konfigurationen

Um einen ersten Vergleich der verschiedenen hier betrachteten Konfigurationen anstellen zu können, wurden Simulationen für alle Konfigurationen über den hier bereits betrachteten Orbit durchgeführt, wobei hier die Berechnungen für die Standard-solarpaneele mit jeweils 14 Solarzellen durchgeführt wurden. Abbildung 9.5 zeigt die Ergebnisse der Verläufe der gesamten erzeugten Energie für die Standard- und AISat-Konfigurationen sowie für die Konfiguration mit um 60° bzw. 90° ausgeklappten Paneelen.

Die in Abbildung 9.3 bereits dargestellte Kurve für die auf der AISat-Konfiguration erzeugten Leistung findet sich darin wieder. Die Unterschiede in den absoluten Werten ergeben sich durch die unterschiedlich angenommene Anzahl der Solarzellen pro Solarpaneel. Man erkennt im Vergleich zur Standardkonfiguration die Abweichungen zu Beginn und am Ende der Sonnenlichtphase aufgrund der Abschattung durch die



**Abbildung 9.5:** Verlauf der gesamten erzeugten Leistungen der verschiedenen Konfigurationen von CLAVIS über einen Umlauf, auf einem Orbit wie bei der AISat-Mission Anfang März

Antennen auf der AISat-Konfiguration. Über einen Zeitraum von gut 30 Minuten sind die Verläufe beider Konfigurationen praktisch identisch. Integriert man die Leistungswerte über den gesamten Orbit, ergibt sich für die Standardkonfiguration ein um ca. 7 % größerer Wert der erzeugten Energie als für die AISat-Konfiguration. Grundsätzlich kann man davon ausgehen, dass die Standardkonfiguration stets etwas mehr Energie erzeugt, als die AISat-Konfiguration, da die Paneelkonstruktionen gleich sind.

Die Konfigurationen mit ausgeklappten Paneelen zeigen grundsätzlich andere Verläufe. Da die Paneele in Richtung des oberen Paneels ausgeklappt werden, ähneln diese Verläufe dem des oberen Paneels in Abbildung 9.3. Diese Verläufe hängen natürlich stark vom gewählten Orbit ab. Wäre die LTDN des SSO 12 h, wären die Werte noch höher, da die Sonne am absteigenden Knoten fast senkrecht auf das obere Paneel schiene, so dass die gesamte Paneelfläche optimal genutzt würde. Befände sich der Satellit mit um 90° ausgeklappten Paneelen in einem Dämmerungsort (LTDN 6 h oder 18 h) mit dem theoretischen Betawinkel von 90°, würde praktisch gar keine Energie erzeugt, da die Einstrahlungsrichtung der Sonne stets parallel zum oberen Paneel läge. Entsprechende Betrachtungen für SSOs finden sich in Abschnitt 10.3.4.

Insgesamt ergibt sich auf diesem Orbit eine um ca. 30 % höhere Energieausbeute der Konfiguration mit um 60° ausgeklappten Paneelen im Vergleich zur Standardkonfiguration. Hier muss jedoch beachtet werden, dass der zeitliche Verlauf der erzeugten Energie ungünstiger ist, als bei der Standardkonfiguration, wo die Gesamtleistung nicht so stark über die Zeit schwankt. Bei den Konfigurationen mit ausklappbaren

Paneele wird der Hauptanteil der Energie in einem relativ kurzen Zeitraum erzeugt, was zur Folge hat, dass sich die Batterie zu Beginn der Sonnenlichtphase stärker entlädt (siehe Abschnitt 9.1.3). Darüber hinaus kann es passieren, dass die maximale Aufladerate der Batterie überschritten wird, wenn die volle Leistung zur Aufladung verwendet wird. Somit wird elektrische Energie in Wärmeenergie dissipiert und damit verschwendet. Außerdem ist es bei diesen Konfigurationen möglich, dass zwei gegenüberliegende Solarpaneele gleichzeitig von der Sonne beschienen werden. Die BCRs, die jeweils an zwei gegenüberliegende Paneele angeschlossen sind, können aber nur insgesamt 8 W aufnehmen (siehe Abschnitt 4.1). Dieser Wert wird überschritten, wenn die Paneele hochgeklappt sind und in günstigem Winkel beschienen werden. Auch dadurch kann wertvolle Energie verloren gehen.

Die beschriebenen Effekte sind bei der Konfiguration mit voll ausgeklappten Paneelen noch stärker. Soll eine dieser Konfigurationen verwendet werden, zum Beispiel auf einem ähnlichen Orbit wie dem der AISat-Mission, muss das Energieversorgungssystem entsprechend angepasst werden.

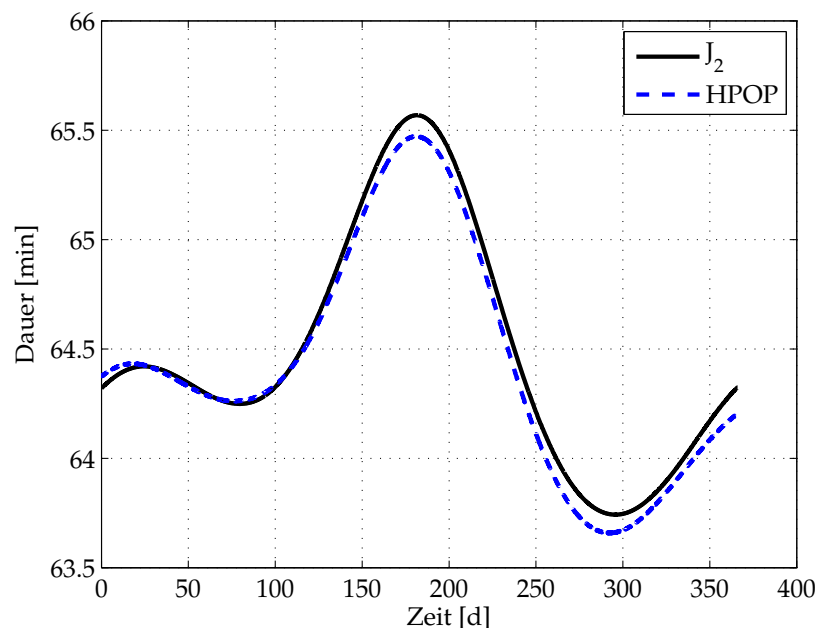
## 9.2 Analyse der Gesamtmission bezüglich der Energieversorgung

Zur Analyse der Gesamtmission der AISat-Konfiguration von CLAVIS werden verschiedene Szenarien herangezogen. Als Startzeitpunkt wird dabei stets der 15. August 2011, 0 h UTC und als Endzeitpunkt der 15. August 2012, 0 h UTC angenommen. Der Orbit wird zu Beginn als ideal kreisförmig in einer Höhe von 650 km mit einer Inklination von  $97,9908^\circ$  definiert. Als Referenzszenario wird der Orbit unter Berücksichtigung der  $J_2$ -Störung propagiert und eine optimale Nadirausrichtung des Satelliten angenommen. Die Ergebnisse werden verglichen mit Ergebnissen für verschieden starke Taumelschwingungen um die Nadirachse und für eine Propagation des Orbits mit dem hochpräzisen Propagator von STK (siehe Abschnitt 8.2.3).

### 9.2.1 Grundlegende Betrachtungen

Zur Betrachtung der während der AISat-Mission erzeugten Energie und der daraus resultierenden Einflüsse auf das DEPS ist zunächst der Verlauf der Dauer der Sonnenlichtperioden über der Missionszeit von Interesse. Abbildung 9.6 zeigt diesen Verlauf, wobei die Werte des HPOP mit denen des  $J_2$ -Propagators verglichen werden.

Zunächst wird deutlich, dass die Dauer der Sonnenlichtperioden auf diesem SSO nur um wenige Minuten im Laufe eines Jahres schwankt. Dabei zeigt sich ein deutliches Maximum im Februar (ca. 180 Tage nach dem 15. August) und ein deutliches Minimum im Juni (ca. 290 Tage nach dem 15. August). Geht man zunächst von einem idealisierten SSO aus, bei dem sich die Orbitalebene stets mit der gleichen Winkelgeschwindigkeit dreht, wie die Erde sich um die Sonne bewegt, würde man das Maximum zur Wintersonnenwende und das Minimum zur Sommersonnenwende erwarten. Da der SSO nicht polar ist, sondern eine Inklination von ca.  $98^\circ$  besitzt, würde in diesem Fall der Betawinkel zur Sommersonnenwende sein Maximum und zur



**Abbildung 9.6:** Verlauf der Dauer der Sonnenlichtperioden während der AISat-Mission, Vergleich zwischen dem  $J_2$ -Propagator und dem HPOP

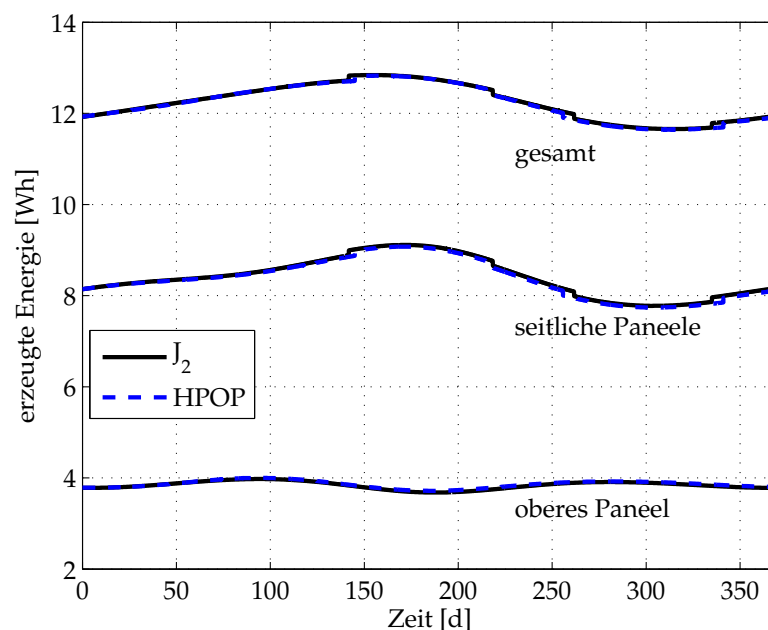
Wintersonnenwende sein Minimum erreichen (siehe auch Abschnitt 3.3.4). Überlagert wird dieser Effekt durch die sich ändernde Bahngeschwindigkeit der Erde. Anfang Januar befindet sich die Erde an ihrem Perihel und hat damit eine verhältnismäßig große Winkelgeschwindigkeit um die Sonne während die Winkelgeschwindigkeit der Drehung der Orbitalebene konstant ist; Anfang Juli zum Aphel-Durchgang ist die Situation gerade umgekehrt. Damit ändert sich die wahre LTDN zu diesen Zeiten geringfügig, was die zeitliche Differenz zwischen den Sonnenwenden und den Extremwerten der Sonnenlichtdauer erklärt.

Der Unterschied zwischen den beiden Kurven in Abbildung 9.6 ergibt sich aus den verschiedenen Funktionsweisen der Propagatoren. Der  $J_2$ -Propagator berücksichtigt nur säkulare Einflüsse der  $J_2$ -Störung, also insbesondere die Drehung der Knotenlinie [21]. Die  $J_2$ -Störung bewirkt jedoch auch eine kurzperiodische Störung der großen Halbachse von einigen Kilometern (siehe auch [6]), so dass die große Halbachse bei dem hier betrachteten Orbit an den Knoten ein Maximum und an den Polen der Bahn ein Minimum erreicht. Definierte man den Orbit von CLAVIS wie gehabt bei einer Anomalie von  $0^\circ$ , würde beim HPOP die Umlaufzeit dadurch insgesamt verringert, was die Ergebnisse verfälschen würde. Daher wird hier der Ausgangsorbit bei einer wahren Anomalie von  $45^\circ$  definiert, so dass der Bahnradius aufgrund der  $J_2$ -Störung um den wahren Wert schwankt und die Umlaufdauern bei beiden Propagatoren zu Beginn der Mission ungefähr gleich sind.

Die mit den beiden verschiedenen Propagatoren berechneten Verläufe der Dauer der Sonnenlichtperioden sind daher zunächst sehr ähnlich. Nach einigen Monaten erkennt man jedoch eine Verringerung der Dauer beim HPOP im Vergleich zum  $J_2$ -Propagator,

die mit der Zeit weiter zunimmt. Dies ist zum einen auf eine geringfügige Absenkung der Bahnhöhe aufgrund atmosphärischen Widerstandes zurückzuführen. Dieser Einfluss ist in der betrachteten Höhe jedoch sehr gering. Wichtiger ist die unterschiedliche Drehrate der Knotenlinie. Aufgrund der genaueren Berücksichtigung der verschiedenen Störungen durch das unregelmäßige Erdgravitationspotential ist die Drehung der Knotenlinie beim HPOP etwas schneller als beim  $J_2$ -Propagator. Dadurch steigt die Differenz des Betawinkels zwischen den beiden Fällen mit der Zeit an, wobei der Winkel beim HPOP geringer ist, was wiederum kürzere Sonnenlichtperioden zur Folge hat (siehe auch Abschnitt 3.2). Der Unterschied in der RAAN beträgt nach einem Jahr wenige Grad.

Der Unterschied in der Sonnenlichtdauer zwischen den beiden verwendeten Propagatoren beträgt jedoch nicht mehr als einige Sekunden und hat keinen allzu großen Einfluss auf die erzeugte Energie. Dies wird in der folgenden Abbildung 9.7 deutlich, in der die jeweils über eine Sonnenlichtphase erzeugte Energie des oberen Paneels, der seitlichen Paneele und die Gesamtenergie als Funktion der Missionsdauer aufgetragen sind.



**Abbildung 9.7:** Verläufe der über eine Sonnenlichtperiode erzeugten Energie des oberen Paneels, der seitlichen Paneele und der Gesamtenergie als Funktion der Missionsdauer für die AISat-Mission mit exakter Nadirausrichtung, Vergleich zwischen dem  $J_2$ -Propagator und dem HPOP

Auf dieser Abbildung zeigt sich zunächst, dass die seitlichen Paneele zusammen ungefähr zwei Drittel der Gesamtenergie liefern und das obere Paneel ungefähr ein Drittel. Die maximalen Abweichungen in der erzeugten Energie zwischen den beiden verschiedenen Propagatoren sind im Bereich von einigen zehntel Wattstunden, was hier maximal 1,1 % entspricht. Da es sich bei diesen Werten sowieso nur um

Abschätzungen handelt, reicht die Analyse mit dem  $J_2$ -Propagator hier aus. Die Verläufe zeigen teilweise geringe, aber sichtbare Sprünge. Diese sind auf die numerische Integration der Leistungen während einer Sonnenlichtphase zu der Gesamtenergie zurückzuführen. Da der dabei verwendete Zeitschritt eine Minute beträgt, entstehen die Sprünge, wenn sich die Dauer der Sonnenlichtphase zum Beispiel von 64,9 min auf 65,0 min erhöht. Im letzteren Fall wird ein Zeitschritt mehr berücksichtigt, was zu dem Sprung führt; in Wirklichkeit sind die Verläufe gleichmäßig.

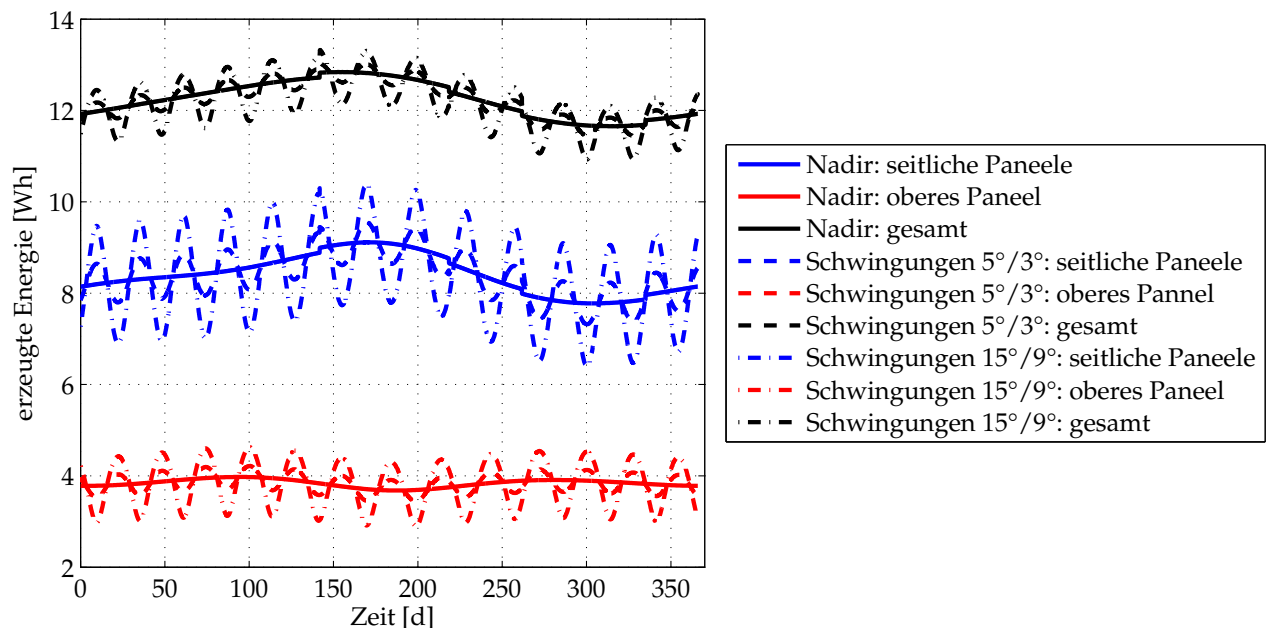
Die Lage der Maxima und Minima im Verlauf der von den seitlichen Paneelen erzeugten Energie entspricht weitgehend dem Verlauf der Sonnenlichtdauer. Dies erklärt sich dadurch, dass bei langer Sonnenlichtdauer der Betawinkel relativ groß ist, was bei der hier vorausgesetzten exakten Nadirausrichtung zur Folge hat, dass die seitlichen Paneele unter einem günstigeren Winkel beschienen werden. Für das obere Paneel verhält es sich umgekehrt. Ist der Betawinkel klein, erzeugt das obere Paneel mehr Energie. In der Summe heben sich diese beiden Effekte ein Stück weit auf, so dass sich die gesamte zur Verfügung stehende Energie über die Mission nur wenig ändert. Ein weiterer Einfluss ist die Änderung der von der Sonne eingestrahlten Leistung pro Fläche über das Jahr (siehe Abschnitt 4.3). Die Extremwerte dieses Verlaufs liegen jedoch ähnlich wie die globalen Extremwerte des Verlaufs der Dauer der Sonnenlichtperioden, sodass beide Effekte einen ähnlichen Einfluss auf die gesamte über eine Sonnenlichtperiode erzeugte Energie besitzen.

Im Mittel liegt diese Energie bei ca. 12 Wh. Geht man von einer durchschnittlichen Leistungsaufnahme von 4,29 W aus (siehe Abschnitt 4.2) ergibt sich unter Berücksichtigung von Verlusten eine benötigte Energie von ca. 7,9 Wh, die problemlos bereitgestellt werden kann. Nach Gleichung (4.12) könnte die Batterie noch voll geladen werden, wenn das Satellitensystem auf Dauer eine Leistung von ca. 6 W aufnimmt.

### 9.2.2 Einfluss der Taumelbewegungen

Wie in Abschnitt 8.1.1 beschrieben, wird CLAVIS in der AISat-Konfiguration aufgrund von Störmomenten Nick- und Rollschwingungen ausführen. Hier werden ein realistisches Szenario mit Nickschwingungen mit einer Amplitude von  $5^\circ$  und Rollschwingungen mit einer Amplitude von  $3^\circ$  und ein Worst-Case-Szenario mit drei Mal größeren Amplituden simuliert, wobei die Dauer einer Schwingung stets der halben Umlaufzeit entspricht. Dabei wurde hier stets der  $J_2$ -Propagator verwendet. Abbildung 9.8 zeigt die Ergebnisse dieser Simulation ähnlich wie Abbildung 9.7 für die verschiedenen Paneelgruppen im Vergleich zur Nadirausrichtung ohne Taumelbewegungen.

Man erkennt bei den Verläufen mit den Taumelbewegungen jeweils eine Art Schwingung um den Verlauf der Energie bei exakter Nadirausrichtung. Auch dabei zeigt sich, dass sich die Abweichungen in den erzeugten Energien der seitlichen Paneele und des oberen Paneels teilweise aufheben. Auch hier gilt grundsätzlich, dass wenn die seitlichen Paneele besonders viel Energie erzeugen, das obere Paneel eher wenig Energie erzeugt. Diese periodischen Abweichungen von der bei Nadirausrichtung erzeugten Energie erfolgen jedoch mit einer deutlich schnelleren Frequenz; eine Periode wird in ungefähr vier Wochen durchlaufen.



**Abbildung 9.8:** Verläufe der über eine Sonnenlichtperiode erzeugten Energie des oberen Paneels, der seitlichen Paneele und der Gesamtenergie als Funktion der Missiondauer für die AISat-Mission, Vergleich zwischen exakter Nadirausrichtung und Taumelbewegungen verschieden großer Amplitude

Dieser Effekt hat zwei Ursachen. Zum einen wird hier die Periode der Taumelschwingungen (zwei komplette Schwingungen pro Orbit) aus der Umlaufdauer des Satelliten berechnet, die wiederum aus der großen Halbachse über Gleichung (3.2) berechnet wird. Diese Gleichung gilt streng genommen jedoch nur für ungestörte Orbits. Auf dem hier unter Berücksichtigung der  $J_2$ -Störung propagierten Orbit ist die Dauer zwischen zwei Äquatordurchgängen jedoch etwas länger. Geht man zunächst näherungsweise davon aus, dass der Satellit stets bei der gleichen wahren Anomalie in das Sonnenlicht eintritt, ändert sich die Orientierung des Satelliten zu diesem Zeitpunkt von Orbit zu Orbit, da die Dauer der zwei Schwingungen um einige Sekunden kürzer ist, als die Umlaufdauer.

Darüber hinaus ist die wahre Anomalie beim Eintritt in das Sonnenlicht nicht konstant, sondern ändert sich aufgrund der Elliptizität des Erdorbits und der damit verbundenen Änderungen der LTDN im Laufe eines Jahres sowie der Änderung der Einstrahlrichtung der Sonne zu den verschiedenen Jahreszeiten. Diese Einflüsse sind dabei nicht konstant über die Zeit, weshalb die Dauer zwischen zwei Maxima der erzeugten Energie ebenfalls nicht konstant ist.

Die Vorhersage der Auswirkungen der Taumelbewegungen ist grundsätzlich nicht trivial. Die Lage der Extremstellen der erzeugten Energie oder die „Phase“ der Schwingung hängt von dem gewählten Ausgangszustand ab. In diesem Fall wird er so gewählt, dass zu Beginn der Mission am aufsteigenden Knoten der Nickwinkel

maximal ist ( $5^\circ$  bzw.  $15^\circ$  in Flugrichtung) und der Rollwinkel gleich null. Ändert man einen dieser Parameter, ergibt sich eine Verschiebung des Verlaufs der erzeugten Energie. Die Dauer zwischen zwei Extremstellen oder die „Frequenz“ ist abhängig von der gewählten Dauer der Taumelbewegungen. Die Frequenz der Nickschwingung beträgt den Simulationen folgend wie bereits erwähnt ungefähr zwei komplette Schwingungen pro Orbit, die Rollschwingung besitzt eine Frequenz in ähnlicher Größenordnung, beide können jedoch nicht exakt vorhergesagt werden.

Aus diesen Gründen ist der hier gezeigte Verlauf in Wirklichkeit nicht genau so zu erwarten. Zeitpunkte der Extremwerte und Zeitspannen dazwischen können hier nicht verlässlich angegeben werden. Jedoch werden in dem gezeigten Szenario regelmäßig Minima und Maxima der erzeugten Energie durchlaufen. Die Differenz zwischen den Werten für den taumelnden und den exakt in Richtung Nadir ausgerichteten Satelliten, also die „Amplitude“ der Schwingung, ist dabei realistisch. Der schlechteste Fall ergibt sich, wenn der Satellit beim Eintritt in die Sonne gerade so geneigt ist, dass die Abschattung der Paneele besonders groß ist, der beste Fall wenn die Abschattung klein ist, die Helixantenne also von der Sonne weg geneigt ist. Die Dauer der Sonnenlichtperioden und die Frequenz der Taumelschwingungen bewirken in der Kombination gerade, dass beim Eintritt in den Erdschatten eine ähnliche Orientierung zur Sonne vorliegt, wie beim Eintritt in die Sonne, was die Ausbildung der Minima und Maxima verstärkt.

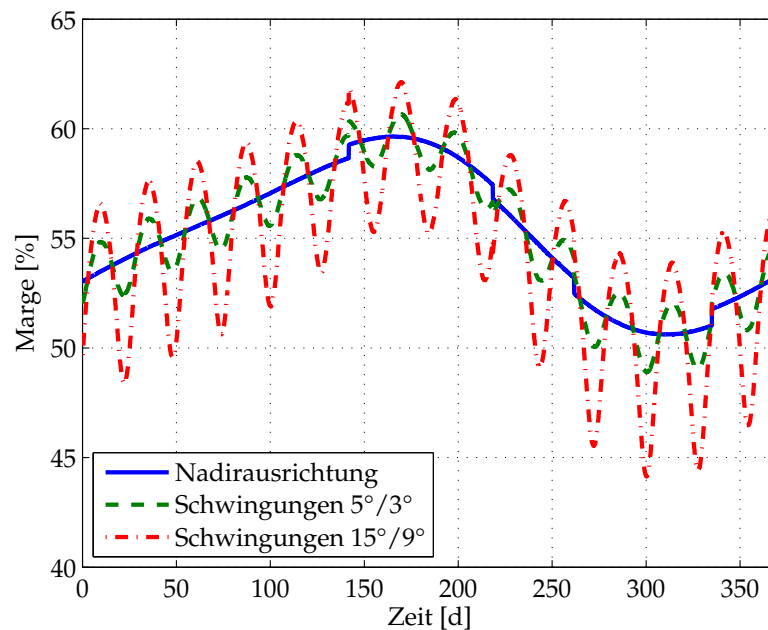
Die Abweichungen in der erzeugten Energie sind bei den seitlichen Paneelen aufgrund der Abschattung relativ groß und heben sich mit den Abweichungen beim oberen Paneel teilweise auf. Insgesamt ergeben sich Abweichungen für den realistischen Fall der Nick- und Rollschwingungen im Bereich von ca.  $\pm 2\%$  und für den Worst Case von  $\pm 5\%$ . Bildet man den Durchschnitt der Abweichungen aufgrund der Taumelbewegungen ergibt sich stets sogar ein positiver Wert. Die Taumelbewegungen führen im Durchschnitt also zu einer Erhöhung der erzeugten Energie, da sich der Einstrahlwinkel auf zwei Paneele gleichzeitig durch Abweichungen von der Nadirrichtung im Bereich des absteigenden Knotens verbessern kann.

### 9.2.3 Parameter der Batterie

Aus den Werten der erzeugten Energie zu jeder Sonnenlichtperiode lassen sich nun die in Abschnitt 4.4 definierten Parameter bestimmen, die die Auf- und Entladung der Batterie begrenzen. Abbildung 9.9 zeigt zunächst die Marge zwischen der erzeugten und der benötigten Energie in Prozent in Abhängigkeit der Missionsdauer für die exakte Nadirausrichtung und die beiden Fälle der Taumelbewegung, wie in Gleichung (4.8) definiert. Zur Berechnung des Orbits wurde der  $J_2$ -Propagator verwendet.

Man erkennt grundsätzlich ähnliche Verläufe wie für die gesamte erzeugte Energie, was auf die zu Grunde liegende Berechnung aus der erzeugten Energie zurückzuführen ist. Die geringste Marge in der erzeugten Energie wird demnach im Juni erreicht, wo dieser bei sehr ungünstiger Taumelbewegung lediglich 44 % betragen kann. Durchschnittlich liegt die Marge jedoch bei ca. 55 % und ist damit stets ausreichend für den nominellen Betrieb des Satelliten. Diese Marge entspricht näherungsweise dem Pro-



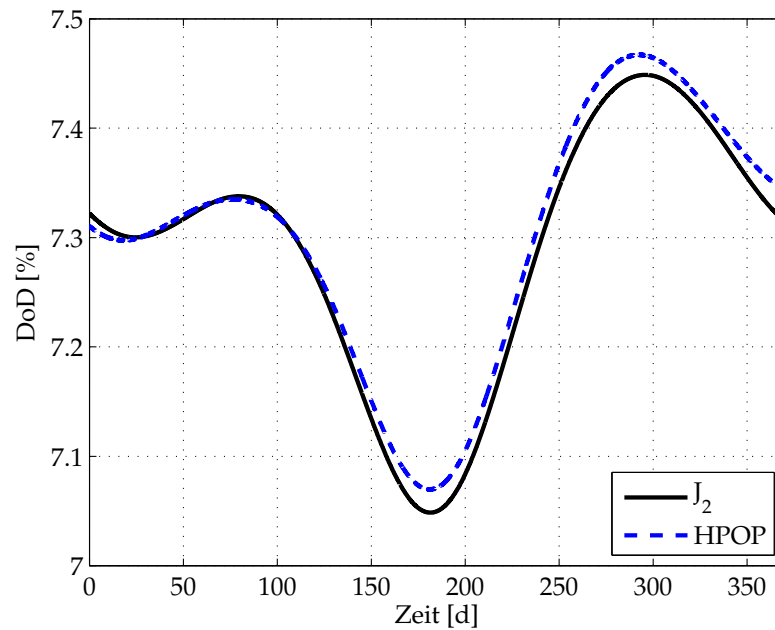


**Abbildung 9.9:** Verläufe der Marge zwischen erzeugter und benötigter Energie als Funktion der Missionsdauer für die AISat-Mission, Vergleich zwischen exakter Nadirausrichtung und Taumelbewegungen verschieden großer Amplitude

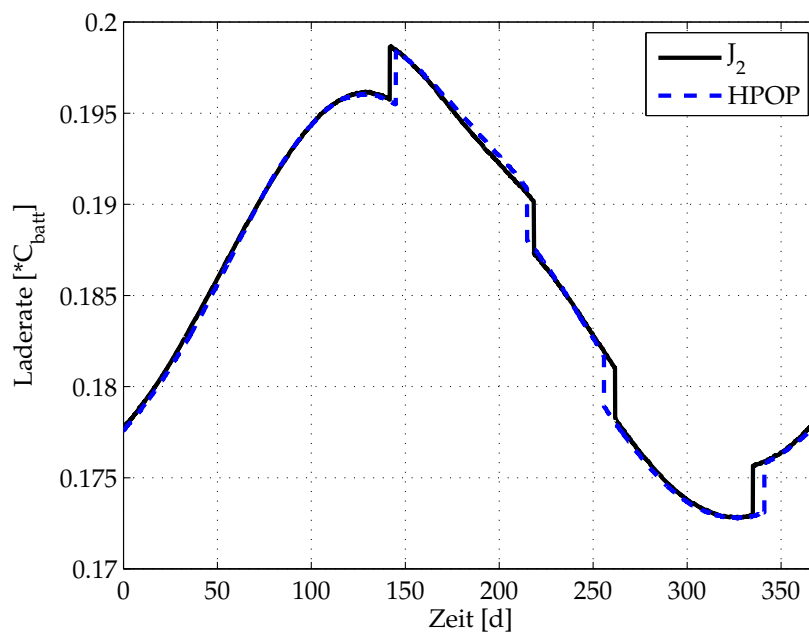
zentsatz der Zeit, die die Batterien des Satelliten während einer Sonnenlichtperiode im voll geladenen Zustand verbringen. Es wird also durchschnittlich nur ca. 45 % der Zeit im Sonnenlicht benötigt, um die Batterien voll zu laden. Der Verlauf unter Verwendung des HPOP liegt maximal ca. 1 % unterhalb des hier dargestellten Verlaufs bei exakter Nadirausrichtung und wurde der Übersichtlichkeit halber hier weggelassen.

Abbildung 9.10 zeigt die Verläufe der DoD in Prozent über die Missionsdauer für die Bahnberechnung mit dem  $J_2$ -Propagator und dem HPOP, Gleichung (4.9) folgend. Die DoD hängt dabei nur von der Dauer der Ekliptik ab, weshalb hier nur der Vergleich zwischen den Propagatoren gezeigt ist. Das Lageverhalten des Satelliten hat keinen Einfluss. Man erkennt daher einen der Dauer der Sonnenlichtzeit entgegengesetzten Verlauf, wobei die maximale DoD bei knapp 7,5 % liegt, deutlich unter dem maximal erlaubten Wert von 20 % (siehe Abschnitt 4.1). Auch hierdurch ergeben sich also keinerlei Einschränkungen.

Abbildung 9.11 zeigt den Verlauf der durchschnittlichen Aufladerate der Batterie als Vielfaches der Batteriekapazität als Funktion der Missionsdauer. Dabei wurde vorausgesetzt, dass die Marge zwischen erzeugter und benötigter Energie dem Prozentsatz der Zeit im Sonnenlicht entspricht, während dem die Batterie bereits voll geladen ist. Die übrige Zeit wurde als Aufladezeit der Batterie angenommen, wobei eine konstante Aufladerate bei konstanter erzeugter Energie angenommen wird. Somit wird die tatsächliche Aufladerate nicht wie in Gleichung (4.10) mit der gesamten Sonnenlichtdauer  $t_{ill}$ , sondern als konservativer Fall mit einer kürzeren Aufladezeit



**Abbildung 9.10:** Verlauf der DoD der Batterie als Funktion der Missionsdauer für die AISat-Mission mit exakter Nadirausrichtung, Vergleich zwischen dem  $J_2$ -Propagator und dem HPOP



**Abbildung 9.11:** Verläufe der Aufladerate der Batterie als Funktion der Missionsdauer für die AISat-Mission mit exakter Nadirausrichtung, Vergleich zwischen dem  $J_2$ -Propagator und dem HPOP

bestimmt. Dadurch ergibt sich eine maximale Aufladerate von knapp  $0,2 \cdot C_{batt}^*$ , was wiederum deutlich unter dem maximal erlaubten Wert von  $1 \cdot C_{batt}^*$  liegt.

Der Wert der Aufladerate steigt mit steigender Dauer der Eklipse und mit steigender Marge zwischen erzeugter und benötigter Energie. Der letztere Einfluss bestimmt hier weitestgehend die Gestalt der Kurve. Die Kombination beider Einflüsse führt dazu, dass der Einfluss der verschiedenen Propagatoren näherungsweise verschwindet, da beim HPOP wie oben beschrieben die Eklipsen länger sind und die Marge der erzeugten Energie geringer ist.

Bei allen in diesem Abschnitt gezeigten Verläufen wird vorausgesetzt, dass der Satellit eine konstante durchschnittliche Leistungsaufnahme besitzt und die Leistung während der Sonnenlichtphase auf konstantem Niveau zur Verfügung steht. Beide Annahmen entsprechen nicht der realen Situation. Ist zum Beispiel der UHF-Transmitter eingeschaltet, steigt über einen relativ kurzen Zeitraum die aufgenommene Leistung relativ stark an, was zu einer erhöhten Entladerate der Batterie führen kann. Eine Überschreitung des Maximalwertes ist dabei aufgrund des BCR nicht möglich. Eine Entladung auf dem Niveau des Maximalwertes über einen längeren Zeitraum sollte hierbei vermieden werden, da die Leistung des DEPS dadurch nachhaltig beeinträchtigt werden kann [9].

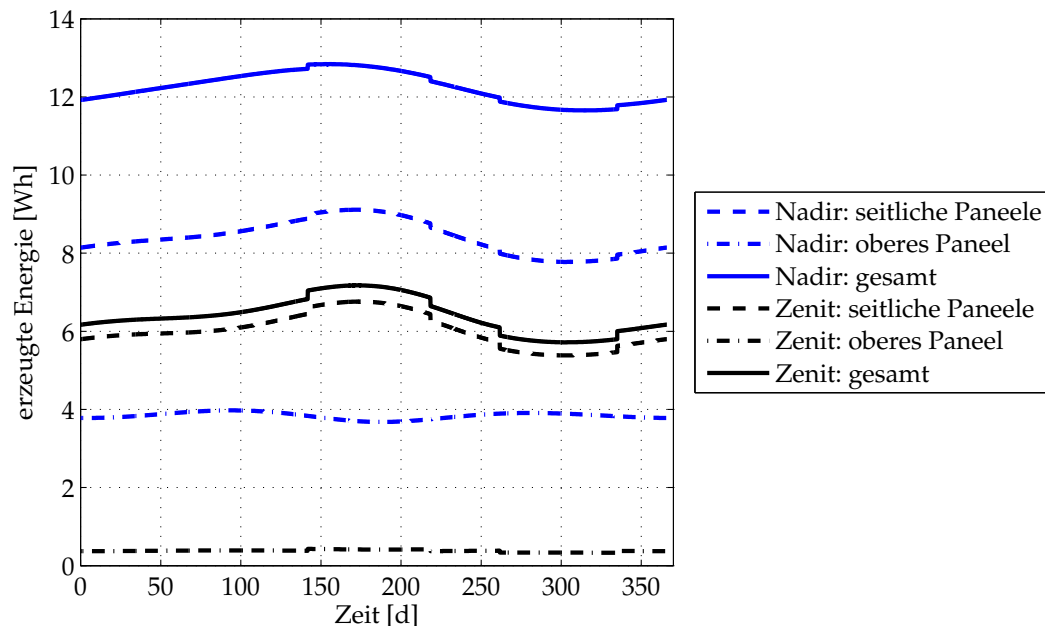
Die Werte für die Marge zwischen erzeugter und benötigter Energie sowie die DoD stellen eine Bilanz eines mittleren Energieverbrauchs dar und können in Wirklichkeit deutlich von den berechneten Werten abweichen. Die vorhandenen Margen sind jedoch relativ groß und eine einmalige Überschreitung der maximal erlaubten Werte (DoD 20 %, Marge der erzeugten Energie 0 %) stellt kein Problem dar, sofern die verbrauchte Energie in der Folgezeit auf entsprechend niedrigerem Niveau gehalten wird.

#### 9.2.4 Zenitausrichtung des Satelliten

Zu Beginn der Mission ist es denkbar, dass sich der Satellit mit der Helixantenne in Richtung Zenit ausrichtet, da Zenit- und Nadirausrichtung energetisch gleichermaßen günstig sind. Eine solche Ausrichtung reduziert die erzeugte Energie drastisch, da das obere Paneel meist durch die Erde abgeschattet wird und die Reflektorplatte sowie die Radialantenne verstärkte Abschattungen der seitlichen Paneele bewirken. Der schlechteste Fall ist dabei die exakte Ausrichtung in Richtung Zenit, die allerdings mit Hilfe der Magnet Torquer vermieden werden kann, so dass sich allenfalls eine Taumbewegung um die Zenitrichtung einstellt. Hier wurde dennoch der Fall der exakten Zenitausrichtung als Worst-Case-Szenario betrachtet.

Abbildung 9.12 zeigt die Verläufe der über die einzelnen Sonnenlichtperioden erzeugten Energie von CLAVIS in der AISat-Konfiguration für die exakte Zenitausrichtung im Vergleich zur exakten Nadirausrichtung, wobei der  $J_2$ -Propagator zur Bahnberechnung verwendet wurde. Ähnlich Abbildung 9.8 sind auch hier die Verläufe für das obere Paneel, die seitlichen Paneele und die Gesamtenergie einzeln aufgeschlüsselt.

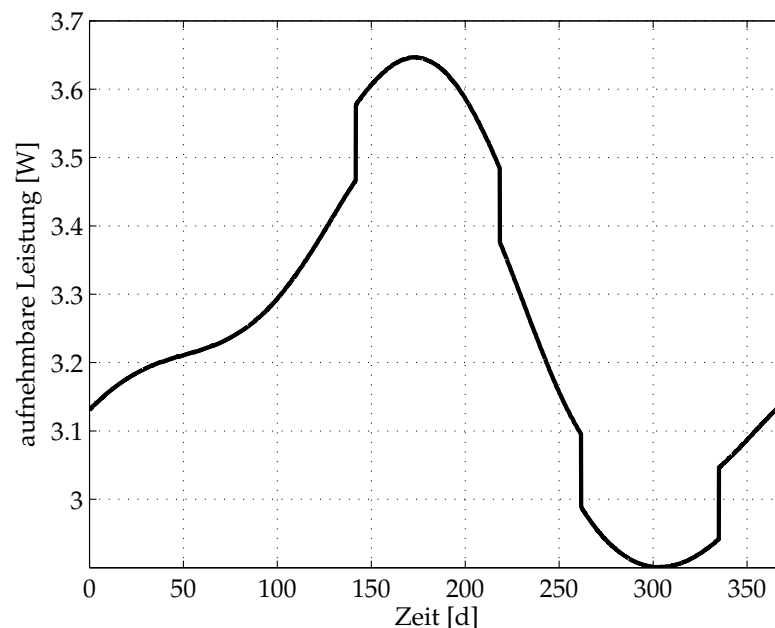
Es zeigt sich, dass das obere Paneel praktisch keine Energie erzeugt während die seitlichen Paneele aufgrund der verstärkten Abschattung deutlich weniger Energie erzeugen, als bei der nominellen Lage. Insgesamt ergibt sich ungefähr eine Halbierung der Energie auf ca. 6 Wh pro Sonnenlichtperiode, wobei die Verläufe über die Zeit denen



**Abbildung 9.12:** Verläufe der über eine Sonnenlichtperiode erzeugten Energie des oberen Paneels, der seitlichen Paneele und der Gesamtenergie als Funktion der Missiondauer für die AISat-Mission, Vergleich zwischen exakter Nadir- und Zenitausrichtung

der Nadirausrichtung stark ähneln. Daraus kann nun wie in Gleichung (4.12) definiert, die maximale Leistung bestimmt werden, die durchschnittlich von dem Satelliten aufgenommen werden kann, so dass die Batterie während jeder Sonnenlichtperiode wieder vollständig geladen wird. Abbildung 9.13 zeigt den Verlauf dieser Leistung über der Missiondauer.

Die Kurve zeigt einen ähnlichen Verlauf wie die Kurve der erzeugten Leistung. Hier ist insbesondere der Beginn der Mission interessant, da eine mögliche Zenitausrichtung des Satelliten zu diesem Zeitpunkt schnellstmöglich behoben werden wird. Dabei ist eine durchschnittliche Leistungsaufnahme von gut 3,1 W möglich. Vergleicht man diesen Wert mit den Werten in Tabelle 4.1, zeigt sich, dass die Leistung ausreicht, um die wichtigsten Instrumente (Computer, DEPS, UHF-Receiver, Batterie- und Telemetrie-elektronik, IMU, Magnet Torquer) unter durchschnittlicher Last zu betreiben. Damit sollte es ohne Probleme möglich sein, den Satelliten in die nominelle Lage zu bringen, zumal hier ein unwahrscheinlicher Worst Case betrachtet wurde. Eine Überschreitung dieser Minimalleistung kann dazu führen, dass die Batterie nicht vollständig geladen wird, was die Kapazität der Batterie zwar beeinträchtigen kann, aber im Einzelfall die Mission nicht gefährdet. DoD und Aufladerate sind bei dieser geringen Leistungsaufnahme geringer als im nominellen Fall.



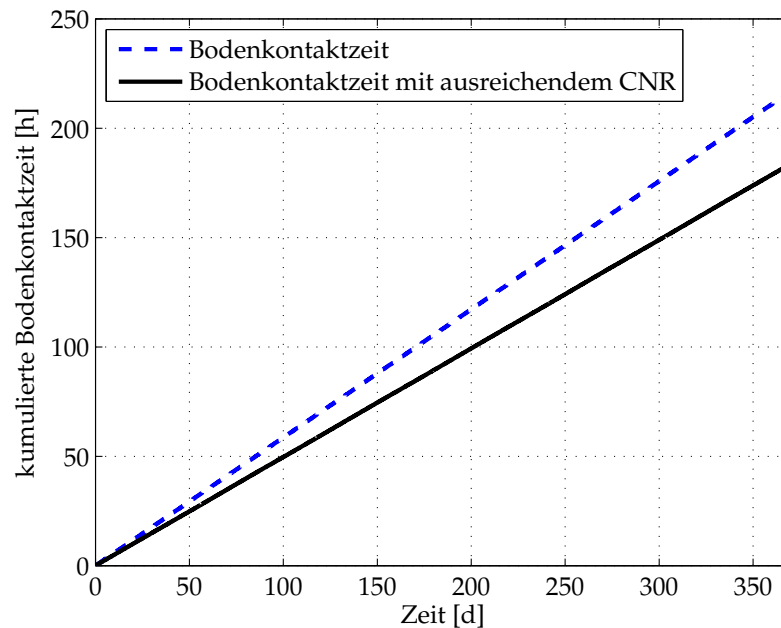
**Abbildung 9.13:** Verlauf der vom Satelliten in Zenitausrichtung maximal aufnehmbaren Leistung, so dass die Batterie in jeder Sonnenlichtphase wieder voll geladen wird

### 9.3 Bodenkontakt während der AISat-Mission

Wie bereits in Abschnitt 8.1.1 erwähnt, wird während der AISat-Mission lediglich die Bodenstation am Institut für Raumfahrtssysteme des DLR in Bremen verwendet. Abbildung 9.14 zeigt die kumulierten Bodenkontaktzeiten für CLAVIS in Abhängigkeit der Zeit nach dem Start der Mission im August sowie die kumulierte Dauer der Bodenkontaktzeiten mit ausreichendem CNR, wobei nur die Funkverbindung vom Satelliten zur Bodenstation betrachtet wird (siehe Abschnitt 5.3). Dabei ist nur dann ein Bodenkontakt vorhanden, wenn die Elevation des Satelliten von der Bodenstation aus gesehen größer ist als die minimale Elevation von  $10^\circ$  (siehe Abschnitt 5.2).

Man erkennt, dass beide Verläufe auf lange Sicht annähernd linear sind, wobei die Bodenkontaktzeit mit ausreichendem CNR im Mittel etwa 15 % kürzer ist, als die Gesamtkontaktzeit. Die einzelnen Bodenkontaktphasen sind dabei unterschiedlich lang, wobei die maximale Länge gut neun Minuten beträgt. Aufgrund der Umlaufdauer von nur gut anderthalb Stunden verteilt sich die Subspur des Satelliten auf der Erdoberfläche über einen Zeitraum von einigen Tagen jedoch gleichmäßig, weshalb sich auf einen längeren Zeitraum betrachtet der lineare Verlauf zeigt. Würde man einen Ausschnitt der Geraden vergrößert betrachten, wären diskrete, unregelmäßige Schritte erkennbar. Durchschnittlich ergibt sich eine Bodenkontaktzeit von ca. 35 min pro Tag und eine Bodenkontaktzeit mit ausreichendem CNR von knapp 30 min pro Tag.

Mit Hilfe der in Abschnitt 5.3 gegebenen Parameter zur Güte der Funkverbindung und den in Abschnitt 5.2 dargelegten Zusammenhängen zur Geometrie der Funk-



**Abbildung 9.14:** Kumulierte Bodenkontaktzeiten und kumulierte Bodenkontaktzeiten mit ausreichendem CNR für die AISat-Mission in Abhängigkeit der Missionzeit

verbindung lässt sich näherungsweise eine Grenzelevation bestimmen, für die das CNR größer ist als der Minimalwert von 10 dB. Dazu setzt man Gleichung (5.13) in Gleichung (5.19) ein und stellt nach der Freiraumdämpfung  $L_{fs}$  um. Setzt man nun die gegebenen, konstanten Verluste sowie 10 dB für das CNR ein, ergibt sich die maximale Freiraumdämpfung, bei der das CNR von 10 dB gerade erreicht wird, zu ca. -150,95 dB. Mit Hilfe von Gleichung (5.12) kann daraus die maximale Schrägentfernung zwischen Satellit und Bodenstation zu ca. 1930,6 km bestimmt werden. Unter Berücksichtigung des Erd- und des Bahnradius kann dann der Kosinussatz auf das linke Dreieck in Abbildung 5.1 angewendet werden, so dass sich die minimale Elevation für ein ausreichendes CNR zu ca.  $11,7^\circ$  ergibt.

Dieser Wert ist jedoch nur als grober Richtwert zu betrachten, da hier eine kugelförmige Erde und ein ideal kreisförmiger Orbit vorausgesetzt werden (siehe auch Abschnitt 5.3.3). Bei einer Bodenstation in höheren Breiten auf der realen abgeplatteten Erde ist die Schrägentfernung zum Satelliten bei geringen Elevationen grundsätzlich größer als es bei einer kugelförmigen Erde der Fall wäre. Damit ergibt sich in Wirklichkeit eine größere minimale Elevation als die berechnete, was den signifikanten Unterschied zwischen den beiden Verläufen in Abbildung 9.14 trotz des geringen rechnerischen Unterschiedes der Elevationen teilweise erklärt. Darüber hinaus spielt jedoch auch die Tatsache eine Rolle, dass Objekte in LEO von einem Betrachter auf der Erdoberfläche aus gesehen in niedrigen Elevationen langsam und in der Nähe des Zenits sehr schnell erscheinen. Daher muss der Großteil der Kommunikation mit der Bodenstation bei niedrigen Elevationen abgewickelt werden, wenn das CNR relativ gering ist. Der Maximalwert des CNR von ca. 19 dB bei hohen Elevationen kann nur wenige

Sekunden lang ausgenutzt werden.

Mit Hilfe von Gleichung (5.10) kann aus den Bodenkontaktzeiten mit ausreichendem CNR die vom Satelliten an die Bodenstation übertragbare Datenmenge berechnet werden. Dazu wird die Zeit zur Initialisierung der Funkverbindung  $t_{init}$  mit 2 s angenommen [16]. Mit dem Faktor  $F_{red}$  wird hier zunächst die Verringerung der übertragenen Datenmenge aufgrund der Kanalkodierung berücksichtigt (siehe Abschnitt 5.1). Bei dem hier verwendeten (255,223)-Reed-Solomon-Code werden 255 Symbole à 8 bit gesendet, von denen 223 Symbole übertragene Daten sind. Die restlichen Symbole dienen der Fehlerkorrektur der eigentlichen Daten, was die Fehlerrate deutlich verringert. Somit wird in  $F_{red}$  aufgrund dieser Kodierung ein Faktor  $\frac{223}{255}$  berücksichtigt. Außerdem werden mit einem Faktor 0,5 Ausfälle der Bodenstation zu Nachtzeiten oder aus anderen Gründen berücksichtigt. Damit ist  $F_{red} = 0,437$ . Es ergibt sich analog zu der Bodenkontaktzeit ein näherungsweise konstanter Wert der übertragbaren Datenmenge pro Zeiteinheit, wenn ein Zeitraum von mindestens einigen Tagen betrachtet wird. Im Mittel beträgt die übertragbare Datenmenge ca. 0,44 MB pro Tag, so dass über die gesamte Mission insgesamt ca. 160 MB vom Satelliten an die Bodenstation übertragen werden können.

# Kapitel 10

## Ergebnisse für zukünftige CLAVIS-Missionen

### 10.1 Näherungsgleichungen

In den folgenden Abschnitten werden stets Näherungsgleichungen für die gezeigten Verläufe angegeben, damit grobe Auslegungen des Energieversorgungssystems für bestimmte Orbits durchgeführt werden können. Dabei werden meist Verläufe betrachtet, die von zwei Variablen abhängen, also grafisch einer Oberfläche entsprechen. In einigen Fällen handelt es sich dabei angenähert um einfache lineare Verläufe, also Ebenen im Raum. Oftmals sind jedoch komplexere Funktionen nötig, um die aus den Simulationen erhaltenen Datensätze mathematisch zu beschreiben. Dabei ist das Hauptproblem, eine geeignete Funktion zu finden, deren Koeffizienten numerisch bestimmt werden können.

Zur Bestimmung dieser Funktionen gibt es verschiedene Softwaretools, so zum Beispiel eine MATLAB-Toolbox oder das Programm EASYFIT, das an der Universität Bayreuth entwickelt wurde [28]. Hier wird jedoch ein Online-Tool verwendet, das im Gegensatz zu den anderen genannten Tools frei zur Verfügung steht. Dabei handelt es sich um die *3D Function Finder*-Funktion auf der Website [zunzun.com](http://zunzun.com) [29]. Als Eingabe benötigt dieses Tool die Daten, die durch eine Funktion angenähert werden sollen, wobei jeder Datenpunkt aus einer x-, y- und z-Koordinate besteht. Es können maximal 10.000 Datenpunkte angegeben werden. Diese Daten werden mit einer großen Anzahl verschiedener Funktionen mit der Methode der kleinsten Quadrate angenähert und die geeignetsten Funktionen aufsteigend nach den gesamten Fehlerwerten sortiert ausgegeben. In einem weiteren Schritt können schließlich die verwendeten Koeffizienten sowie verschiedene Diagramme angezeigt werden.

Auch mit der am Institut für Raumfahrtssysteme des DLR zur Verfügung stehenden Grundversion von MATLAB können gegebene Funktionen den Daten angenähert werden. Dazu werden die Quadrate der Differenzen zwischen den Daten und der angenäherten Funktion mit Hilfe der Funktion *fminsearch* minimiert. Ein Beispiel hierzu ist in [26] gegeben. Die aus dem oben beschriebenen Online-Tool erhaltene Funktion wird unter Verwendung der Koeffizienten als Startwerte mit dieser Methode in MATLAB angenähert, da teilweise mehr als 10.000 Datenpunkte vorliegen. Außer-



dem können somit entsprechende Diagramme in MATLAB erstellt werden. Die hier verwendeten Skripte und Funktionen finden sich im Anhang F bzw. auf der CD zur Arbeit.

Die angegebenen Näherungsgleichungen können bei einer groben ersten Auslegung zur Abschätzung der Sonnenlichtdauer bzw. der erzeugten Energie dienen. Sie gelten jedoch immer nur für kreisförmige Bahnen in dem hier betrachteten Bahnhöhenbereich zwischen 200 km und 650 km sowie für den jeweils angegebenen Inklinationsbereich und die gerade betrachtete Konfiguration.

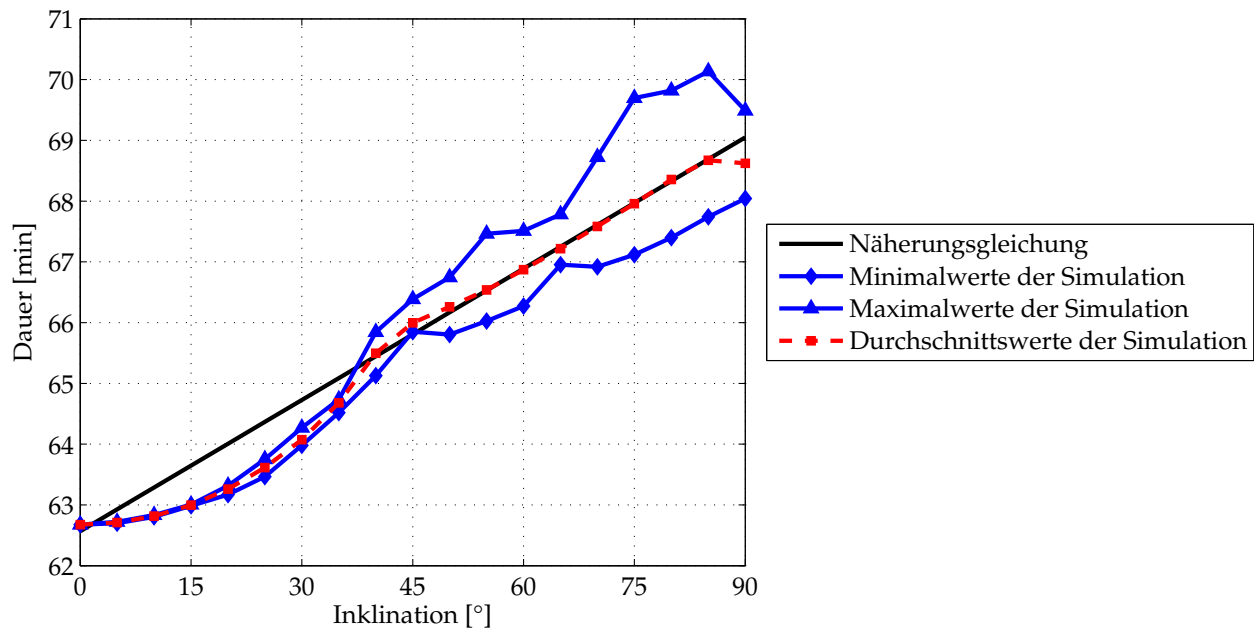
## 10.2 Sonnenlichtzeiten

In diesem Abschnitt werden zunächst Ergebnisse für die Sonnenlichtzeiten vorgestellt, wobei stets die maximale, minimale und durchschnittliche Dauer für verschiedene Orbits betrachtet werden. Die maximale Dauer ist dabei stets die Dauer der längsten Sonnenlichtperiode, die kürzer ist als die Umlaufdauer des Satelliten. Längere Sonnenlichtzeiten betragen dann meist mehrere Tage bis Wochen und sind unkritisch in Bezug auf die erzeugte Energie. Der mittlere Wert ist das arithmetische Mittel aus allen Dauern der Sonnenlichtperioden, die kürzer sind als die Umlaufdauer. Damit wird in bestimmten Fällen, insbesondere auf Dämmerungsortbits, ein signifikanter Zeitraum der Mission über ein Jahr praktisch nicht betrachtet. Während dieses Zeitraumes ist der Satellit jedoch dauerhaft in der Sonne, so dass kein Problem mit der erzeugten Energie auftritt. Die gegebenen Werte gelten also bei Dämmerungsortbits streng genommen nur für den Zeitabschnitt, während dem überhaupt Eklipsen auftreten (siehe auch Abbildung 3.9). Es werden die Ergebnisse für Orbits mit Inklinationen zwischen  $0^\circ$  und  $90^\circ$  vorgestellt und anschließend auf die SSOs eingegangen. Zunächst folgt jedoch eine Einführung in die Art der Darstellung der Ergebnisse in den folgenden Abschnitten.

### 10.2.1 Darstellung der Ergebnisse

Zur Erläuterung der Art der Darstellung der Ergebnisse in den folgenden Abschnitten zeigt Abbildung 10.1 verschiedene Verläufe der durchschnittlichen Sonnenlichtdauer als Funktion der Inklination zwischen  $0^\circ$  und  $90^\circ$  bei einer Bahnhöhe von 650 km. Als Grundlage dient die Simulation der Sonnenlichtzeiten für die entsprechende Bahnhöhe für alle betrachteten Startzeitpunkte und -rektaszensionen. Aus diesen einigen 100 Fällen sind hier die Verläufe der Maximal- und Minimalwerte sowie des arithmetischen Mittels aller Werte über die Inklination dargestellt. Zusätzlich ist die in Abschnitt 10.2.2 beschriebene in diesem Fall lineare Näherung gezeigt.

Die Abbildung zeigt zunächst, dass die mittlere Sonnenlichtdauer für alle Startzeitpunkte und -rektaszensionen in einem relativ schmalen Band liegt. Da hier die Durchschnitts- bzw. Extremwerte über eine Mission über ein Jahr betrachtet werden, ergeben sich bei Inklinationen unter  $90^\circ$  zumeist Verläufe, die nur unwesentlich von Startzeitpunkt und -rektaszension abhängen. Es fällt jedoch auf, dass die Differenz zwischen Minimal- und Maximalwerten der Simulationen mit zunehmender Inklination zunimmt. Dieser Effekt spielt auch bei der Betrachtung der erzeugten



**Abbildung 10.1:** Verläufe der durchschnittlichen Sonnenlichtdauer als Funktion der Inklination auf einer Bahnhöhe von 650 km auf Grundlage der Simulationen verschiedener Startzeitpunkte und -rektaszensionen und der in Abschnitt 10.2.2 gegebenen Näherungsgleichung

Energie eine Rolle und ist auf die  $J_2$ -Störung zurückzuführen. Die Geschwindigkeit der Knotenliniendrehung aufgrund dieser Störung lässt mit zunehmender Inklination nach und verschwindet auf polaren Orbits. Im Laufe eines Jahres werden auf den hochinklinierten Orbits die möglichen Extremfälle des Betawinkels nicht in jedem Fall durchlaufen, was sich in der mittleren Dauer der Sonnenlichtzeiten widerspiegelt. Die langsamere Drehung der Knotenlinie verringert die Änderung des Betawinkels mit der Zeit, so dass der Betawinkel zu Beginn der Mission und damit Startzeitpunkt und -rektaszension einen merkbaren Einfluss haben. Auf niedriger inklinierten Bahnen tritt dieser Effekt aufgrund der schnelleren Knotenliniendrehung nicht auf. Auch bei der minimalen Sonnenlichtdauer fällt der Effekt nicht ins Gewicht, da die maximale Eklipsendauer wie in Abbildung 3.8 gezeigt auch auf hochinklinierten Orbits während eines Jahres mehrfach auftritt.

Da die Abweichungen der Dauer der Sonnenlichtperioden für die verschiedenen Startzeitpunkte und -rektaszensionen relativ gering sind (einige Minuten), wird diese Abhängigkeit hier nicht genauer betrachtet. Zur Näherung der Verläufe mit geeigneten Näherungsgleichungen werden jedoch die Werte aller simulierten Szenarien herangezogen. Damit die Genauigkeit der Näherungsgleichungen abgeschätzt werden kann, wird zu jeder Gleichung der durchschnittliche und der maximale Fehler in Prozent angegeben. Beim maximalen Fehler handelt es sich um den Betrag der maximalen Abweichung zwischen einem Wert aus der Simulation und der Näherung bezogen auf den Wert aus der Simulation. Zur Berechnung des durchschnittlichen Fehlers wird für jeden Wert aus den Simulationen der auf diesen Wert bezogene Betrag der Abweichung zwischen Näherung und Simulationswert bestimmt und der

arithmetische Mittelwert aus all diesen Fehlerwerten berechnet.

In den folgenden Abschnitten werden zumeist Verläufe über zwei Variablen betrachtet: Bei Inklinationen zwischen  $0^\circ$  und  $90^\circ$  sind dies die Bahnhöhe und die Inklination und bei SSOs die LTDN und die Bahnhöhe, aus der sich wiederum die Inklination nach Gleichung (3.10) ergibt. Um trotz der nötigen dreidimensionalen Darstellung die Aussagekraft der Diagramme nicht zu gefährden, wird bei Verläufen für Inklinationen unter  $90^\circ$  jeweils nur das arithmetische Mittel aller Werte für die verschiedenen hier betrachteten Bahnhöhen und Inklinationen gezeigt (siehe auch Abschnitt 8.1.2). Darüber hinaus sind im Anhang G für alle dreidimensionalen Diagramme entsprechende zweidimensionale Diagramme wiedergegeben, auf denen die Werte besser abgelesen werden können.

Für SSOs ist dieses Vorgehen nicht praktikabel, da bei der Simulation Startzeitpunkt und RAAN variiert werden, während die Dauer der Sonnenlichtzeiten eine Funktion der LTDN ist, die sich aus diesen beiden Werten ergibt. Aufgrund der gewählten Startwerte ergibt sich jedoch für jeden simulierten Fall eine unterschiedliche LTDN, weshalb nicht einfach ein Mittelwert gebildet werden kann. Da die Unterschiede in den Sonnenlichtzeiten auf den SSOs aufgrund der näherungsweise konstanten Orientierung der Orbits zur Sonne jedoch deutlich geringer sind als auf niedriger inklinierten Orbits, werden hier alle zur Verfügung stehenden Werte in den Diagrammen geplottet.

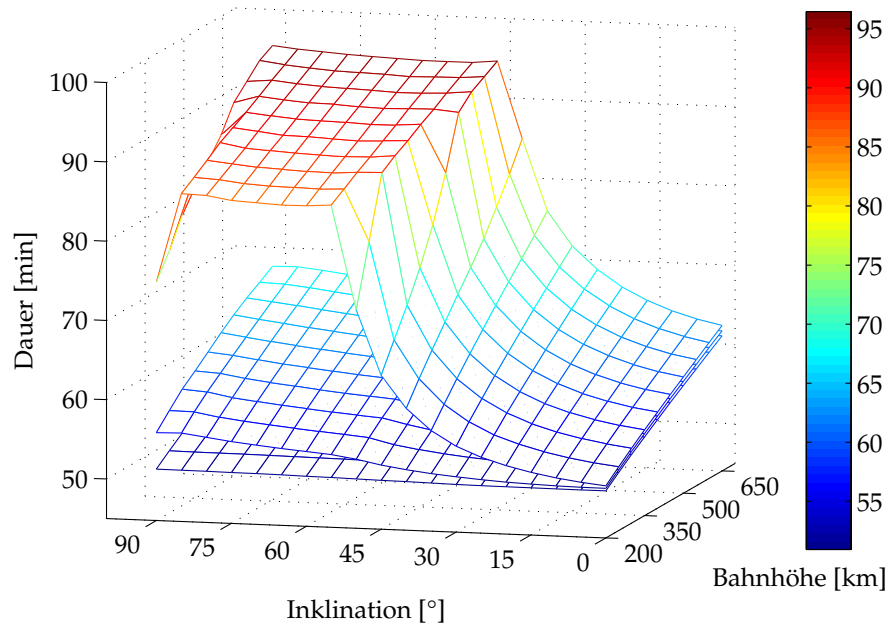
In allen Fällen wird dazu die MATLAB-Funktion *mesh* verwendet, mit der der Verlauf der Funktionswerte einer Funktion zweier Variablen durch ein farbiges Gitter dargestellt wird. Die Farbe ist dabei abhängig von der z-Koordinate, in diesem Fall also der Dauer der Sonnenlichtperiode bzw. der erzeugten Energie. Mit Hilfe des stets dargestellten Farbbalkens lassen sich entsprechende Werte ablesen.

### 10.2.2 Orbits mit Inklinationen zwischen $0^\circ$ und $90^\circ$

Abbildung 10.2 zeigt die Verläufe von kürzester, mittlerer und längster Sonnenlichtdauer als Funktion von Bahnhöhe und Inklination über ein Jahr, wobei der Mittelwert der Simulationen der verschiedenen Startzeitpunkte und -rektaszensionen dargestellt ist.

Die Minimale Sonnenlichtdauer ist dabei primär eine Funktion der Bahnhöhe. Wie in Abbildung 3.3 gezeigt wurde, hängt die maximale Eklipsendauer unter den zunächst getroffenen Annahmen ausschließlich von der Bahnhöhe ab. Die minimale Sonnenlichtdauer kann daraus mit Hilfe von Gleichung (3.2) als Differenz zwischen Umlaufdauer und maximaler Eklipsendauer berechnet werden. Der resultierende Verlauf ist aufgrund der gegenläufigen Trends beider Größen näherungsweise linear.

Berücksichtigt man, dass die Erde keine Kugel ist, ergibt sich auch eine geringe Abhängigkeit der minimalen Sonnenlichtdauer von der Inklination. Auf hochinklinierten Orbits sind die kürzesten Sonnenlichtperioden etwas länger als auf äquatornahen Bahnen, da hier die Abplattung der Erde eine kürzere Schattenperiode zur Folge hat. Der Unterschied beträgt maximal etwa eine halbe Minute und verläuft ebenfalls nä-



**Abbildung 10.2:** Verläufe von kürzester, mittlerer und längster Sonnenlichtdauer als Funktion von Bahnhöhe und Inklination, Mittelwerte über alle Startzeitpunkte und -rektaszensionen, siehe auch Abbildung G.1

herungsweise linear mit der Inklination. Der Verlauf der minimalen Sonnenlichtdauer als Funktion von Bahnhöhe und Inklination kann also mit einer einfachen linearen Funktion der Form

$$t_{ill}(h, i) = q_1 + q_2 \cdot h + q_3 \cdot i \quad (10.1)$$

angenähert werden. Die an die simulierten Werte anzupassenden Faktoren werden im Rahmen dieser Arbeit stets mit  $q$  bezeichnet. Für den hier betrachteten Fall von Inklinationen zwischen  $0^\circ$  und  $90^\circ$  und Bahnhöhen zwischen 200 km und 650 km sind die Faktoren in Tabelle 10.1 gegeben, wobei die Bahnhöhe in Kilometern und die Inklination in Grad eingesetzt werden müssen und sich die Dauer in Minuten ergibt. Man erkennt anhand der Größenordnung der Faktoren, dass der Einfluss der Inklination hier gering ist. Der maximale dabei auftretende Fehler ist ca. 0,5 %, im Durchschnitt beträgt er 0,15 %.

$q_1 =$	$4,6192 \cdot 10^1$	[min]
$q_2 =$	$2,4479 \cdot 10^{-2}$	[min/km]
$q_3 =$	$6,3792 \cdot 10^{-3}$	[min/°]

**Tabelle 10.1:** Faktoren der Näherungsgleichung (10.1) zur Abschätzung der minimalen Sonnenlichtdauer als Funktion von Bahnhöhe und Inklination für Inklinationen zwischen  $0^\circ$  und  $90^\circ$

Der Verlauf der mittleren Sonnenlichtdauer folgt dem der minimalen Sonnenlichtdauer bei einer Inklination von  $0^\circ$  recht genau; hier hat nur die sich ändernde

Einstrahlungsrichtung der Sonne übers Jahr einen Einfluss. Mit steigender Inklination steigt die mittlere Sonnenlichtdauer an, jedoch deutlich weniger stark als die maximale Dauer. Dies kann mit Abbildung 3.6 erklärt werden. Für den dabei betrachteten Orbit mit einer Inklination von  $50^\circ$  ist die Dauer der Eklipsen meist relativ lang, also in der Nähe des Maximums und wird nur selten deutlich kürzer. Analog dazu sind die Sonnenlichtperioden meist nicht deutlich länger als die minimale Sonnenlichtdauer, so dass der Anstieg der mittleren Dauer mit der Inklination relativ gering ist.

Insgesamt ergibt sich bei den durchschnittlichen Sonnenlichtzeiten ebenfalls ein annähernd linearer Verlauf über Inklination und Bahnhöhe, so dass auch hier Gleichung (10.1) zur Annäherung herangezogen wird. Es ergeben sich die in Tabelle 10.2 aufgeführten Faktoren, wobei ein maximaler Fehler 3 % auftritt, der durch die Abweichungen bei hohen Inklinationen zustande kommt (siehe auch Abbildung 10.1). Der durchschnittliche Fehler liegt bei ca. 0,6 %.

$q_1 =$	$4,4988 \cdot 10^1$	[min]
$q_2 =$	$2,7043 \cdot 10^{-2}$	[min/km]
$q_3 =$	$7,2046 \cdot 10^{-2}$	[min/°]

**Tabelle 10.2:** Faktoren der Näherungsgleichung (10.1) zur Abschätzung der durchschnittlichen Sonnenlichtdauer als Funktion von Bahnhöhe und Inklination für Inklinationen zwischen  $0^\circ$  und  $90^\circ$

Die maximale Sonnenlichtdauer steigt, nachdem sie bei äquatorialen Orbits wiederum kaum von der mittleren Dauer abweicht, stark mit der Inklination an, da der größtmögliche Betawinkel dabei ebenfalls stark ansteigt. Oberhalb einer Grenzinklination treten Sonnenlichtzeiten auf, deren Dauer größer ist als ein Orbit und die hier nicht berücksichtigt werden, weshalb die maximale Dauer dabei grundsätzlich der Umlaufdauer nahe kommt. Die aus der fehlenden  $J_2$ -Störung resultierenden Abweichungen bei polaren Orbits werden später genauer betrachtet. Die Grenzinklination  $i_{\text{grenz}}$  ergibt sich als Funktion der Bahnhöhe unter Verwendung von Gleichung (3.5) zu

$$i_{\text{grenz}} = \beta_{\text{grenz}} - i_{\ddot{A}q,Ekl} = \arcsin\left(\frac{R_E}{R_E + h}\right) - i_{\ddot{A}q,Ekl} \quad (10.2)$$

Der Verlauf der maximalen Dauer der Sonnenlichtzeiten bei Inklinationen kleiner als  $i_{\text{grenz}}$  lässt sich grob durch einen Quotienten zweier Polynome beschreiben, bei höheren Inklinationen wird der Wert auf die Umlaufdauer gesetzt. Es ist also

$$t_{\text{ill}}(h, i) = \begin{cases} \frac{q_1 + q_2 \cdot h + q_3 \cdot i + q_4 \cdot h \cdot i}{1 + q_5 \cdot h + q_6 \cdot i + q_7 \cdot h \cdot i}, & i < i_{\text{grenz}}(h) \\ t_U(h), & i \geq i_{\text{grenz}}(h) \end{cases} \quad (10.3)$$

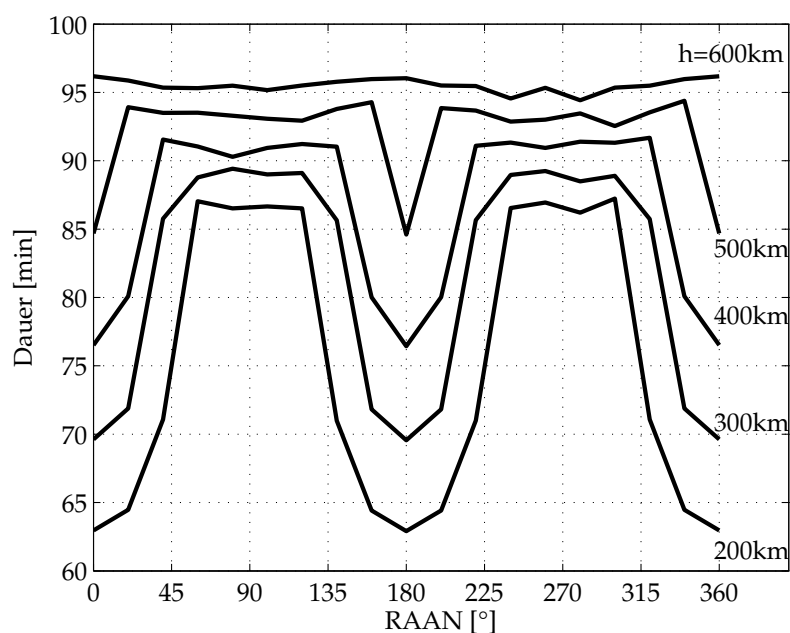
Als beste mit der Methode der kleinsten Fehlerquadrate berechnete Näherung für die Faktoren ergeben sich die in Tabelle 10.3 aufgeführten Werte. Dabei ist der maximale Fehler der gesamten Funktion ca. 10,7 %, ein relativ hoher Wert, der kurz unter der Grenzinklination auftreten kann, da sich die Dauer der Sonnenlichtperiode hier stark ändert, so dass geringe Schwankungen in Startzeitpunkt und -rektaszension zu verhältnismäßig großen Unterschieden der Sonnenlichtdauer bei konstanter Bahnhöhe und Inklination führen können. Der durchschnittliche Fehler liegt bei ca. 2 %.

$q_1 =$	$4,9188 \cdot 10^1$	[min]	$q_5 =$	$- 3,7069 \cdot 10^{-4}$	[1/km]
$q_2 =$	$- 3,4363 \cdot 10^{-3}$	[min/km]	$q_6 =$	$- 1,9565 \cdot 10^{-2}$	[1/°]
$q_3 =$	$- 1,0113$	[min/°]	$q_7 =$	$1,5255 \cdot 10^{-5}$	[1/(° km)]
$q_4 =$	$9,5055 \cdot 10^{-4}$	[min/(° km)]			

**Tabelle 10.3:** Faktoren der Näherungsgleichung (10.3) zur Abschätzung der maximalen Sonnenlichtdauer als Funktion von Bahnhöhe und Inklination für Inklinationen zwischen  $0^\circ$  und  $i_{\text{grenz}}$

### Maximale Sonnenlichtdauer auf polaren Orbits

Wie in Abbildung 10.2 zu erkennen, sinkt der Durchschnittswert der maximalen Sonnenlichtdauer bei einer Inklination von  $90^\circ$  geringfügig. Bei dieser Inklination ist die  $J_2$ -Störung nicht vorhanden, so dass die über die Zeit konstante RAAN einen Einfluss hat, wobei das Startdatum unerheblich ist. Abbildung 10.3 zeigt die Verläufe der maximalen Sonnenlichtdauer als Funktion der RAAN für verschiedene Bahnhöhen.



**Abbildung 10.3:** Verläufe der längsten Sonnenlichtperioden für polare Orbits als Funktion der RAAN für verschiedene Bahnhöhen

Man erkennt zunächst Unterschiede in der maximalen Sonnenlichtdauer von mehr als 20 min auf niedrigen Orbits, die oberhalb einer Bahnhöhe von 600 km praktisch verschwinden. Auf niedrigen Orbits zeigt sich darüber hinaus, dass die maximale Sonnenlichtdauer, die kürzer als die Umlaufdauer ist, bei einer RAAN von  $90^\circ$  oder  $270^\circ$  ungefähr der Umlaufdauer entspricht, während sie bei RAAN von  $0^\circ$  und  $180^\circ$  minimal wird. Dieser Unterschied ergibt sich durch die unterschiedliche Orientierung der Bahnebene zu den Tagundnachtgleichen. Ist die RAAN  $90^\circ$  oder  $270^\circ$ , beträgt der Betawinkel zu den Tagundnachtgleichen  $90^\circ$ , so dass im Laufe eines Jahres mindestens zwei Sonnenlichtperioden auftreten, die deutlich länger sind, als

die Umlaufdauer. Ist die RAAN  $0^\circ$  und  $180^\circ$ , wird der Betawinkel zu den Sonnenwenden maximal, kann aber nie einen Wert von  $90^\circ - 23,5^\circ = 66,5^\circ$  überschreiten, da die Sonne zu den Sonnenwenden in einem Winkel von  $23,5^\circ$  zur Äquatorebene auf die Erde scheint. Dieser maximale Betawinkel ist kleiner als der Grenzfall des Betawinkels für Bahnhöhen unter 577 km (siehe Abschnitt 3.2), so dass auf diesen niedrigen Bahnen keine Sonnenlichtperiode auftritt, die länger ist als die Umlaufdauer.

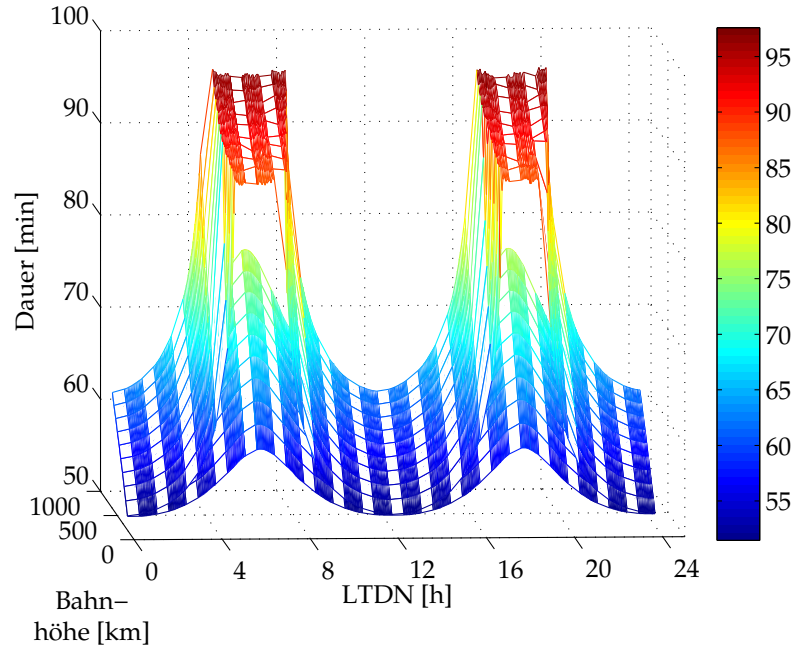
Die hier dargestellten Verläufe beinhalten die längsten in den Simulationen auftretenden Sonnenlichtperioden, die kürzer sind als die rechnerische Umlaufdauer. Es zeigt sich auf niedrigen Umlaufbahnen, dass die Sonnenlichtdauer den Maximalwert nicht bei den theoretisch zu erwartenden RAAN von  $90^\circ$  bzw.  $270^\circ$  erreicht, sondern in der Nähe dieser Werte längere Sonnenlichtzeiten erreicht werden. Dies erklärt sich damit, dass die Sonnenlichtdauer keinen stetigen Verlauf über die Zeit besitzt, sondern jedem Orbit ein diskreter Wert zugeordnet wird. Betrachtet man nun die Sonnenlichtdauer einen Orbit bevor eine Sonnenlichtperiode über mehrere Orbits erreicht wird, so ist diese Sonnenlichtperiode etwas kürzer als die Umlaufdauer und ist damit eine mögliche Sonnenlichtphase maximaler Dauer nach dem hier angelegten Kriterium. Die genaue Dauer ist jedoch abhängig vom gewählten Startorbit sowie der Propagation und muss nicht bei der RAAN, bei der die maximale Sonnenlichtdauer zu erwarten ist, maximal werden. Daher treten gewisse Schwankungen der maximalen Sonnenlichtdauer in diesen Bereichen der RAAN auf.

### 10.2.3 Sonnensynchrone Orbits

Auf kreisförmigen sonnensynchronen Orbits wird die Inklination direkt durch die Bahnhöhe bestimmt, weshalb zunächst nur die Bahnhöhe als Einflussfaktor berücksichtigt wird. Darüber hinaus spielt die eingangs eingeführte LTDN eine wichtige Rolle, da sie den Betawinkel und damit die Dauer der Sonnenlichtperioden bestimmt. Abbildung 10.4 zeigt den Verlauf von minimaler und maximaler Sonnenlichtdauer als Funktion der Bahnhöhe und der LTDN. Der Verlauf der mittleren Werte wird der Übersichtlichkeit halber zunächst weggelassen. Auch hier werden sämtliche Startzeitpunkte und -rektaszensionen berücksichtigt. Zur Berechnung der LTDN aus diesen Werten wird das MATLAB-Skript *start\_LTDN.m* verwendet, welches im Anhang A.1 wiedergegeben ist.

Dabei zeigt sich zunächst, dass die Verläufe im Bereich der LTDN zwischen 0 h und 12 h und im Bereich zwischen 12 h und 24 h praktisch identisch sind, weshalb hier im Folgenden nur noch der Bereich zwischen 0 h und 12 h betrachtet wird. Außerdem erkennt man eine ähnliche lineare Abhängigkeit der Sonnenlichtdauer von der Bahnhöhe wie bei den niedriger inklinierten Orbits. Darüber hinaus zeigt sich, dass nur sehr geringe Abweichungen aufgrund unterschiedlicher Startzeitpunkte vorhanden sind, da die Verläufe ansonsten Sprünge aufweisen würden.

Die Verläufe über die LTDN sind erwartungsgemäß. Auf Dämmerungsorts, also bei einer LTDN von ca. 6 h oder 18 h, ist der Betawinkel stets relativ groß, was zu Sonnenlichtzeiten führt, die länger sind als ein Umlauf. Daher existiert auch hier ein Bereich, in dem die längste Sonnenlichtdauer im Rahmen der getroffenen Annahmen



**Abbildung 10.4:** Verläufe der kürzesten und längsten Sonnenlichtperioden für SSOs als Funktion der LTDN und der Bahnhöhe, siehe auch Abbildung G.2

ungefähr der Umlaufdauer entspricht. Auch die minimale Sonnenlichtdauer liegt bei Dämmerungsorts deutlich über der minimal möglichen für die jeweilige Bahnhöhe. Auf SSOs mit einer LTDN im Bereich von 0h bzw. 12h entsprechen maximale und minimale Sonnenlichtdauer hingegen ungefähr der minimalen Dauer für die betrachtete Bahnhöhe, da hier der Betawinkel bei ungefähr  $0^\circ$  relativ konstant ist (siehe auch Abschnitt 3.3.4).

Die Verläufe für die minimale Sonnenlichtdauer für eine LTDN zwischen 0h und 12h können relativ genau mit zwei Funktionen angenähert werden, die jeweils Kombinationen von Exponentialfunktionen sind. Für die Sonnenlichtdauer gilt näherungsweise

$$t_{ill}(h, LTDN) = \begin{cases} q_1 + q_2 \cdot Z_1 + q_3 \cdot Z_2 + q_4 \cdot Z_1^2 + q_5 \cdot Z_2^2 + q_6 \cdot Z_1 \cdot Z_2, & 0h \leq LTDN < 6h \\ q_{11} + q_{12} \cdot Z_3 + q_{13} \cdot Z_4 + q_{14} \cdot Z_3^2 + q_{15} \cdot Z_4^2 + q_{16} \cdot Z_3 \cdot Z_4, & 6h \leq LTDN \leq 12h \end{cases} \quad (10.4)$$

mit

$$\begin{aligned} Z_1 &= \exp(q_7 \cdot h + q_8) \\ Z_2 &= \exp(q_9 \cdot LTDN + q_{10}) \\ Z_3 &= \exp(q_{17} \cdot h + q_{18}) \\ Z_4 &= \exp(q_{19} \cdot LTDN + q_{20}) \end{aligned}$$

Die darin vorkommenden Faktoren für die minimale Sonnenlichtdauer sind in Tabelle 10.4 zusammengefasst, wobei die LTDN in Stunden und die Bahnhöhe in Kilometern



in die Gleichung eingesetzt werden muss. Der maximale Fehler bei dieser Näherung tritt an der Schnittstelle zwischen beiden Funktionen bei einer LTDN von 6 h auf und beträgt ca. 1,7 %; der durchschnittliche Fehler liegt lediglich bei ca. 0,2 %.

$q_1 =$	$1,0624 \cdot 10^2$	[min]	$q_{11} =$	$1,0778 \cdot 10^2$	[min]
$q_2 =$	$- 1,6488 \cdot 10^3$	[min]	$q_{12} =$	$- 5,1244 \cdot 10^3$	[min]
$q_3 =$	$7,8886 \cdot 10^2$	[min]	$q_{13} =$	$2,3292 \cdot 10^2$	[min]
$q_4 =$	$1,0918 \cdot 10^3$	[min]	$q_{14} =$	$5,6639 \cdot 10^2$	[min]
$q_5 =$	$- 1,5836 \cdot 10^3$	[min]	$q_{15} =$	$- 1,4178 \cdot 10^2$	[min]
$q_6 =$	$- 1,6383 \cdot 10^4$	[min]	$q_{16} =$	$- 1,4829 \cdot 10^4$	[min]
$q_7 =$	$- 4,9754 \cdot 10^{-4}$	[1/km]	$q_{17} =$	$- 4,7724 \cdot 10^{-4}$	[1/km]
$q_8 =$	$- 3,2649$	[-]	$q_{18} =$	$- 4,3948$	[-]
$q_9 =$	$0,6354$	[1/h]	$q_{19} =$	$- 0,6390$	[1/h]
$q_{10} =$	$- 6,2035$	[-]	$q_{20} =$	$2,6838$	[-]

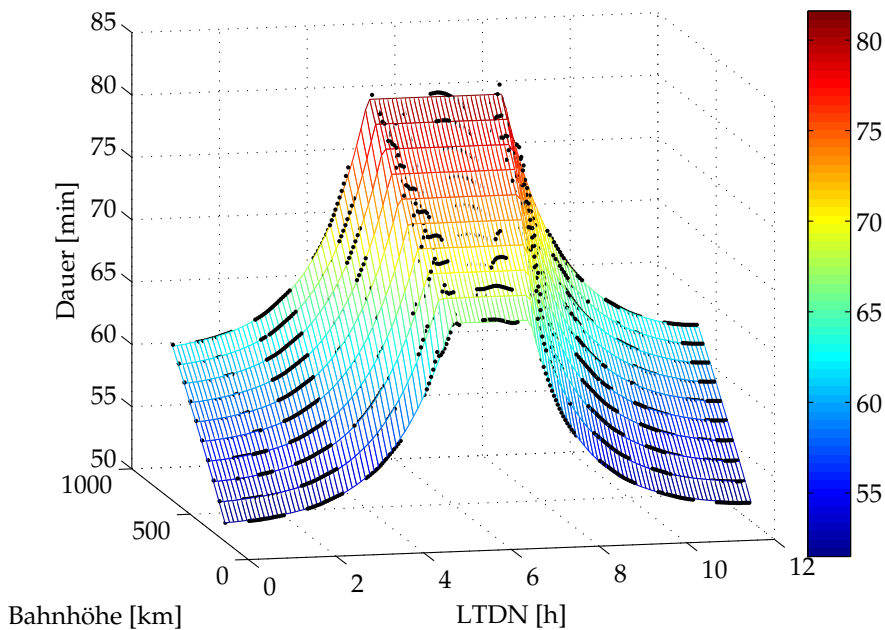
**Tabelle 10.4:** Faktoren der Näherungsgleichung (10.4) zur Abschätzung der minimalen Sonnenlichtdauer als Funktion von Bahnhöhe und LTDN für SSOs

Für die maximale Sonnenlichtdauer kann ebenfalls Gleichung (10.4) zur Annäherung des Verlaufs herangezogen werden. Dabei ist jedoch zu beachten, dass um eine LTDN von 6 h herum die Funktion nicht stetig ist. Die maximale Sonnenlichtdauer wird hier auf die Umlaufdauer gesetzt, sofern das Ergebnis der Näherungsgleichung größer ist als die Umlaufdauer. Der Bereich der LTDN, in dem diese Ersetzung notwendig ist, kann nicht einfach angegeben werden, da er von der Bahnhöhe abhängt; auf höheren Bahnen treten Orbits ohne Eklipse eher auf, als auf niedrigeren. Die für die Abschätzung der maximalen Sonnenlichtdauer verwendeten Parameter sind in Tabelle 10.5 gegeben. Der maximale bei dieser Abschätzung auftretende Fehler tritt im Übergangsbereich zwischen der Exponentialfunktion und der Definition der Sonnenlichtdauer als der Umlaufdauer (bei einer LTDN von ungefähr 4 h bzw. 7 h) auf und beträgt ca. 8,3 %. Der durchschnittliche Fehler liegt bei ca. 0,7 %.

$q_1 =$	$7,9533 \cdot 10^1$	[min]	$q_{11} =$	$1,2537 \cdot 10^2$	[min]
$q_2 =$	$- 3,7961 \cdot 10^3$	[min]	$q_{12} =$	$- 3,0251 \cdot 10^3$	[min]
$q_3 =$	$4,8162 \cdot 10^2$	[min]	$q_{13} =$	$5,8607 \cdot 10^3$	[min]
$q_4 =$	$1,1814 \cdot 10^5$	[min]	$q_{14} =$	$2,0117 \cdot 10^4$	[min]
$q_5 =$	$6,4708 \cdot 10^2$	[min]	$q_{15} =$	$1,0233 \cdot 10^4$	[min]
$q_6 =$	$- 3,5934 \cdot 10^4$	[min]	$q_{16} =$	$- 1,8045 \cdot 10^5$	[min]
$q_7 =$	$- 1,5609 \cdot 10^{-3}$	[1/km]	$q_{17} =$	$- 4,2917 \cdot 10^{-4}$	[1/km]
$q_8 =$	$- 4,1957$	[-]	$q_{18} =$	$- 3,4079$	[-]
$q_9 =$	$9,3204 \cdot 10^{-1}$	[1/h]	$q_{19} =$	$- 9,7172 \cdot 10^{-1}$	[1/h]
$q_{10} =$	$- 6,3946$	[-]	$q_{20} =$	$3,6464$	[-]

**Tabelle 10.5:** Faktoren der Näherungsgleichung (10.4) zur Abschätzung der maximalen Sonnenlichtdauer als Funktion von Bahnhöhe und LTDN für SSOs, auf denen keine Umläufe ohne Eklipse auftreten

Für die durchschnittlichen Sonnenlichtdauern weisen die Verläufe grundsätzlich eine ähnliche Struktur auf wie die Verläufe der maximalen Sonnenlichtdauer, besitzen also auch einen Bereich der LTDN um 6 h herum, in dem die durchschnittliche Dauer bei gleichbleibender Bahnhöhe nicht weiter steigt. Dies ist darauf zurückzuführen, dass der Mittelwert der Dauer der Sonnenlichtzeiten mit den maximalen Zeitspannen in der Sonne, die kürzer sind als die Umlaufzeit, berechnet wird. Abbildung 10.5 zeigt den Verlauf der durchschnittlichen Sonnenlichtdauer als Funktion der Bahnhöhe und der LTDN, wobei die Punkte den simulierten Werten entsprechen und der farbige Verlauf der Näherung.



**Abbildung 10.5:** Verläufe der Sonnenlichtperioden durchschnittlicher Dauer für SSOs als Funktion der Bahnhöhe und der LTDN, Simulationsergebnisse als Punkte und angenäherter Verlauf, siehe auch Abbildung G.2

Wie bei dem Verlauf der maximalen Sonnenlichtdauer erkennt man auch hier, dass die Grenzen des Bereichs der LTDN, in dem die Sonnenlichtdauer konstant ist, von der Bahnhöhe abhängt. In diesem Fall wurde eine lineare Funktion gesucht, die die maximale durchschnittliche Sonnenlichtdauer als Funktion der Bahnhöhe beschreibt, da die Simulationsergebnisse näherungsweise einen linearen Verlauf zeigen. Überschreitet die durch die Näherungsgleichung berechnete durchschnittliche Sonnenlichtdauer diesen Grenzwert  $t_{ill,grenz}$ , wird die Dauer auf diesen Wert gesetzt. Für den Grenzwert wurde aus den Simulationsergebnissen die folgende Bestimmungsgleichung gefunden:

$$t_{ill,grenz} = 0,032889 \frac{\text{min}}{\text{km}} \cdot h + 60,222 \text{ min} \quad (10.5)$$

Die Koeffizienten der Näherungsgleichungen zur Abschätzung der durchschnittlichen Sonnenlichtdauer auf SSOs sind in Tabelle 10.6 zusammengefasst. Der maximale bei

der Näherung auftretende Fehler ist ca. 4,2 % und resultiert aus der Fluktuation der Simulationswerte in dem Bereich, in dem die durchschnittliche Sonnenlichtdauer auf einer Bahnhöhe als konstant angenommen wird. Der mittlere Fehler beträgt ca. 0,4 %.

$q_1 =$	$1,2233 \cdot 10^2$	[min]	$q_{11} =$	$9,6250 \cdot 10^1$	[min]
$q_2 =$	$- 1,1708 \cdot 10^4$	[min]	$q_{12} =$	$- 5,5498 \cdot 10^3$	[min]
$q_3 =$	$5,3829 \cdot 10^1$	[min]	$q_{13} =$	$6,7226 \cdot 10^3$	[min]
$q_4 =$	$- 3,4143 \cdot 10^4$	[min]	$q_{14} =$	$1,0210 \cdot 10^5$	[min]
$q_5 =$	$- 3,7809 \cdot 10^{-1}$	[min]	$q_{15} =$	$2,5373 \cdot 10^4$	[min]
$q_6 =$	$- 7,8996 \cdot 10^3$	[min]	$q_{16} =$	$- 5,5832 \cdot 10^5$	[min]
$q_7 =$	$- 3,5531 \cdot 10^{-4}$	[1/km]	$q_{17} =$	$- 7,1947 \cdot 10^{-4}$	[1/km]
$q_8 =$	$- 5,0504$	[-]	$q_{18} =$	$- 4,4699$	[-]
$q_9 =$	$8,4404 \cdot 10^{-1}$	[1/h]	$q_{19} =$	$- 7,7117 \cdot 10^{-1}$	[1/h]
$q_{10} =$	$- 3,3517$	[-]	$q_{20} =$	$8,2687 \cdot 10^{-1}$	[-]

**Tabelle 10.6:** Faktoren der Näherungsgleichung (10.4) zur Abschätzung der durchschnittlichen Sonnenlichtdauer als Funktion von Bahnhöhe und LTDN für SSOs für Sonnenlichtdauern unter  $t_{ill,grenz}$

Mit den hier angegebenen Näherungsgleichungen lassen sich die minimalen, maximalen und durchschnittlichen Sonnenlichtzeiten auf SSOs abschätzen. Dabei befinden sich die Faktoren  $q_7$  bis  $q_{10}$  bzw.  $q_{17}$  bis  $q_{20}$  jeweils im Argument der Exponentialfunktion und bewirken, dass das Ergebnis der Funktion nicht über die Maßen groß oder klein wird. Ihre Einheiten sind so gewählt, dass das Argument der Exponentialfunktion dimensionslos wird. Die anderen Faktoren wichten die Einflüsse der verschiedenen linearen und quadratischen Kombinationen der Exponentialfunktionen und besitzen als Einheit daher Minuten.

## 10.3 Erzeugte Energie als Funktion der Sonnenlichtdauer

### 10.3.1 Grundlegende Betrachtungen

Bei der Auswertung der aus den Simulationen gewonnen Daten zur erzeugten Energie zeigte sich für die betrachteten Konfigurationen ein direkter Zusammenhang zwischen der Dauer der Sonnenlichtperiode und der erzeugten Energie bei einer bestimmten Bahnhöhe, wobei kein anderer Faktor, insbesondere auch nicht die Inklination, einen merklichen Einfluss hat. Dies ist darauf zurückzuführen, dass der Betawinkel, der die Dauer der Sonnenlichtperiode bestimmt, bei der hier angenommenen Nadirausrichtung des Satelliten ebenfalls die Einstrahlrichtung auf den Satelliten bestimmt. Betrachtet man zunächst die Standardkonfiguration wird bei einem Betawinkel nahe  $90^\circ$  das obere Paneel praktisch nicht beschienen, die seitlichen Paneele liefern die komplette Energie. Ist der Betawinkel ungefähr  $0^\circ$ , so ist der Einstrahlwinkel auf das obere Paneel in der Nähe des in der Sonne liegenden Knotens sehr günstig, so dass

dieses viel Energie erzeugt, während die seitlichen Paneele nur einen geringeren Beitrag liefern. In der Konfiguration mit ausklappbaren Paneelen ähnelt die Ausrichtung der seitlichen Paneele zur Sonne je nach Ausklappwinkel der des oberen Paneels (siehe Abbildung 2.2). Diese Konfigurationen erzeugen daher während einer kurzen Sonnenlichtperiode mehr Energie als während einer längeren. Die Bahnhöhe bewirkt dabei einen Unterschied der Betawinkel für zwei unterschiedlich hohe Orbits bei gleicher Sonnenlichtdauer.

In den folgenden Abschnitten werden diese Zusammenhänge genauer dargestellt und mathematisch beschrieben. Um für diese Untersuchungen eine breite Datenbasis zur Verfügung zu haben, wird eine weitere Simulation durchgeführt, bei der für verschiedene Inklinationen und Bahnhöhen bei einer bestimmten RAAN und einem bestimmten Startzeitpunkt die von den verschiedenen Konfigurationen während aller Sonnenlichtperioden eines Jahres erzeugte Energie bestimmt wird. Die Bahnhöhe wird dabei wie gehabt in 50 km-Schritten zwischen 200 km und 650 km Höhe variiert, während die Inklination zwischen  $0^\circ$  und  $90^\circ$  in  $15^\circ$ -Schritten variiert wird. Das dazu verwendete Skript sowie die zur Auswertung der Daten verwendeten Skripte finden sich im Anhang F.

Für diese Simulation wird stets von einer exakten Ausrichtung des Satelliten in Richtung Nadir ausgegangen. Der Einfluss von Taumelbewegungen kann mit den in Abschnitt 9.2.2 für die AISat-Konfiguration gegebenen Werten abgeschätzt werden. Aufgrund der bei den nun betrachteten Konfigurationen nicht vorhandenen Abschattung durch Antennen bzw. die Reflektorplatte stellen diese Werte eine konservative Annahme dar.

Die hier gegebenen Näherungsgleichungen können dazu verwendet werden, die von CLAVIS erzeugte Energie bei gegebener Dauer einer Sonnenlichtperiode abzuschätzen. Dabei können verschiedene Änderungen der Solarpaneele und der von der Sonne eingestrahlten Leistung berücksichtigt werden, wie im nächsten Abschnitt gezeigt wird.

### 10.3.2 Gestalt der Näherungsgleichungen

Die genaue Anordnung der Solarzellen auf den einzelnen Solarpaneelen bei zukünftigen CLAVIS-Missionen steht zu diesem Zeitpunkt noch nicht genau fest. Die in Abbildung 4.1 gezeigte Anordnung nutzt den vorhandenen Platz nicht optimal aus und auch bei der Größe der Paneele ist in der Standardkonfiguration wie in Abbildung 2.1 dargestellt noch Spielraum. Da dieser Spielraum jedoch unterschiedlich ausfällt, ist es auf zukünftigen Missionen auch denkbar, dass das obere Paneel und die seitlichen Paneele unterschiedlich viele Solarzellen besitzen. Für die Konfiguration mit ausklappbaren Paneelen wird derzeit eine Version geprüft, bei der auf dem oberen Paneel 28 Solarzellen und auf den seitlichen Paneelen jeweils 24 Solarzellen vorgesehen sind [2].

Um diese Unsicherheiten bei der Simulation der Daten zu umgehen, wird stets angenommen, dass die in Abbildung 4.1 gezeigte Standard-Anordnung mit 14 Solarzellen

verwendet wird. Der hier verwendete Weg zur Berechnung der erzeugten Energie (siehe Abschnitt 8.3.2) erlaubt es, eine unterschiedliche Anzahl an Solarzellen pro Paneel zu berücksichtigen, indem die erzeugte Energie mit dem Faktor  $\frac{n_{cell}}{14}$  multipliziert wird. Darin ist  $n_{cell}$  die Anzahl der Solarzellen auf dem betrachteten Paneel, die auf die bei der Simulation verwendete Anzahl von 14 Solarzellen bezogen wird. Damit kann jede unterschiedliche Anzahl an Solarzellen auf den verschiedenen Paneelen berücksichtigt werden.

Darüber hinaus sind auch Konfigurationen mit weniger als vier seitlichen Paneelen denkbar, um Platz für Nutzlasten zu schaffen. Wie in Abschnitt 8.2.1 gezeigt wird, kann dieser Einfluss mit dem Faktor  $\frac{n_{pan,side}}{4}$  berücksichtigt werden, wobei  $n_{pan,side} = [1..4]$  die Anzahl der tatsächlich vorhandenen seitlichen Paneele angibt. Außerdem wird bei den Simulationen das Minimum der Sonneneinstrahlung nach Gleichung (4.2) von  $1325,8 \frac{W}{m^2}$  verwendet. Auch dieser Wert kann bei gegebener Jahreszeit mit dem Faktor  $\frac{P_{\odot}}{1325,8 W/m^2}$  angepasst werden.

Für die Bestimmung der Näherungsgleichungen wird jeweils das obere Paneel getrennt von den seitlichen Paneelen betrachtet, da verschiedene Anzahlen an Solarzellen angedacht sind und die Berechnung für das obere Paneel bei allen Konfigurationen gleich ist. Grundsätzlich gilt also, dass die gesamte über eine Sonnenlichtperiode erzeugte Energie  $E$  der Summe der von allen seitlichen Paneelen zusammen erzeugten Energie  $E_{side}$  und der vom oberen Paneel erzeugten Energie  $E_{top}$  unter Berücksichtigung der genannten Faktoren entspricht. Es ist

$$E = \frac{P_{\odot}}{1325,8 \frac{W}{m^2}} \cdot \left( \frac{n_{cell,side}}{14} \cdot \frac{n_{pan,side}}{4} \cdot E_{side} + \frac{n_{cell,top}}{14} \cdot E_{top} \right) \quad (10.6)$$

Bei der Konfiguration mit um  $90^\circ$  ausgeklappten Paneelen muss dabei beachtet werden, dass die durch die seitlichen Paneele erzeugte Energie über die vom oberen Paneel erzeugte Energie berechnet wird. Diese gilt jedoch, anders als bei den anderen Konfigurationen, nur für ein einzelnes Paneel, weshalb der Vorfaktor bei den seitlichen Paneelen angepasst werden muss. Darüber hinaus wird dabei stets die gleiche Energie durch alle Paneele erzeugt, so dass  $E_{side} = E_{top} = E_{pan,90}$  gilt. Damit ergibt sich

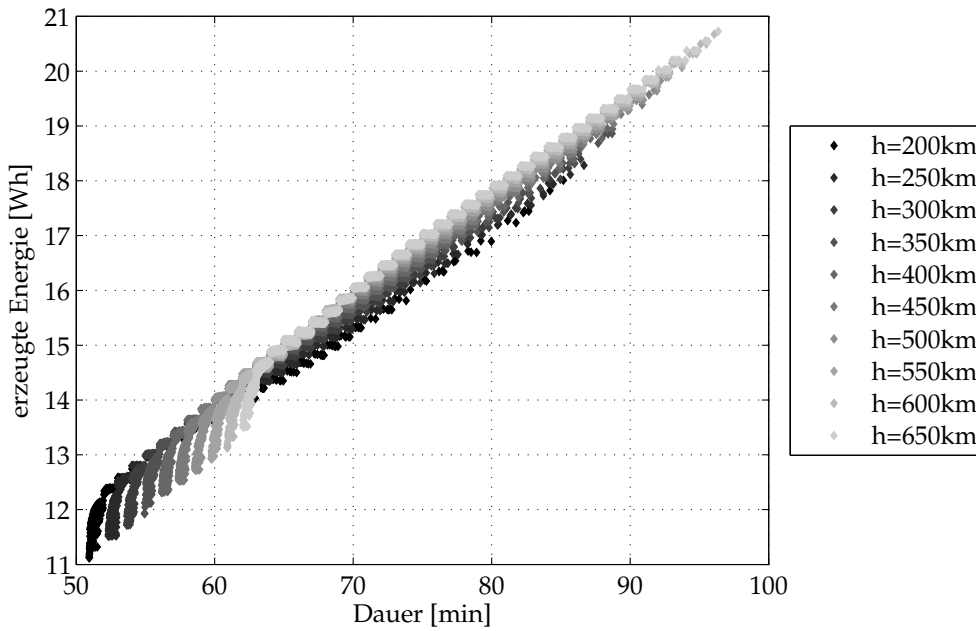
$$E_{90} = E_{pan,90} \cdot \frac{P_{\odot}}{1325,8 \frac{W}{m^2}} \cdot \left( \frac{n_{cell,side}}{14} \cdot n_{pan,side} + \frac{n_{cell,top}}{14} \right) \quad (10.7)$$

Die im Folgenden gezeigten Diagramme gelten jeweils für den Fall mit allen vier seitlichen Paneelen mit jeweils 14 Solarzellen pro Paneel am Minimum der von der Sonne eingestrahnten Leistung, um eine Vergleichbarkeit zu gewährleisten. Die auf diese Weise berechnete Energie kann dann in Gleichung (4.12) eingesetzt werden, um die maximale Leistungsaufnahme des Satelliten abzuschätzen, bei der die Batterie stets wieder komplett geladen wird.

### 10.3.3 Standardkonfiguration

Abbildung 10.6 zeigt ein Streudiagramm der erzeugten Energie der Standardkonfiguration über der Dauer der Sonnenlichtperiode für die verschiedenen Bahnhöhen. Man

erkennt einen näherungsweise linearen Anstieg der erzeugten Energie mit der Dauer, wobei die Bahnhöhe ebenfalls einen Einfluss hat. Für eine gegebene Bahnhöhe zeigt sich ein relativ genau definierter Verlauf der erzeugten Energie über der Dauer der Sonnenlichtperiode. Grundsätzlich ist hier zu erkennen, dass die erzeugte Energie bei gleicher Sonnenlichtdauer mit steigender Bahnhöhe zunimmt, wie in Abschnitt 10.3.1 beschrieben.



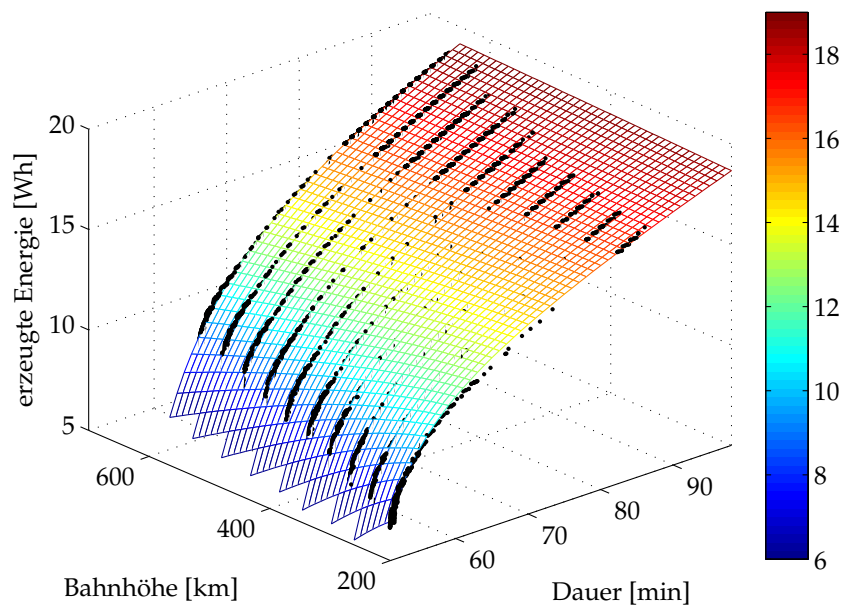
**Abbildung 10.6:** Streudiagramm der von der Standardkonfiguration erzeugten Energie als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen

Abbildung 10.7 zeigt die Verläufe der erzeugten Energie der seitlichen Paneele als Funktion der Dauer der Sonnenlichtperiode und der Bahnhöhe. Darin sind die Punkte die Simulationsergebnisse und der farbige Verlauf entspricht der Näherungsgleichung. Man erkennt, dass die Energie ähnlich wie in Abbildung 10.6 mit der Dauer ansteigt. Da die seitlichen Paneele zusammen im Allgemeinen mehr Energie erzeugen als das obere Paneel, wird der Verlauf insbesondere bei langen Sonnenlichtperioden durch die seitlichen Paneele bestimmt. Bei kurzen Sonnenlichtperioden erzeugt das obere Paneel wie in Abschnitt 10.3.1 dargestellt verhältnismäßig viel und die seitlichen Paneele verhältnismäßig wenig Energie.

Die dabei verwendete Näherungsgleichung hat die in diesem Abschnitt stets verwendete Form

$$E_{side/top}(t_{ill}, h) = \frac{q_1 + q_2 \cdot t_{ill} + q_3 \cdot h + q_4 \cdot t_{ill} \cdot h}{1 + q_5 \cdot \ln(t_{ill}) + q_6 \cdot \ln(h) + q_7 \cdot \ln(t_{ill}) \cdot \ln(h)} \quad (10.8)$$

Die für die Berechnung der durch die seitlichen Paneele erzeugten Energie verwendeten Faktoren sind in Tabelle 10.7 zusammengefasst. Dabei muss beachtet werden, dass



**Abbildung 10.7:** Verlauf der erzeugten Energie aller seitlichen Paneele der Standardkonfiguration als Funktion der Dauer der Sonnenlichtperiode und der Bahnhöhe, Simulationsergebnisse als Punkte und Verlauf nach der Näherungsgleichung (10.8), siehe auch Abbildung G.3

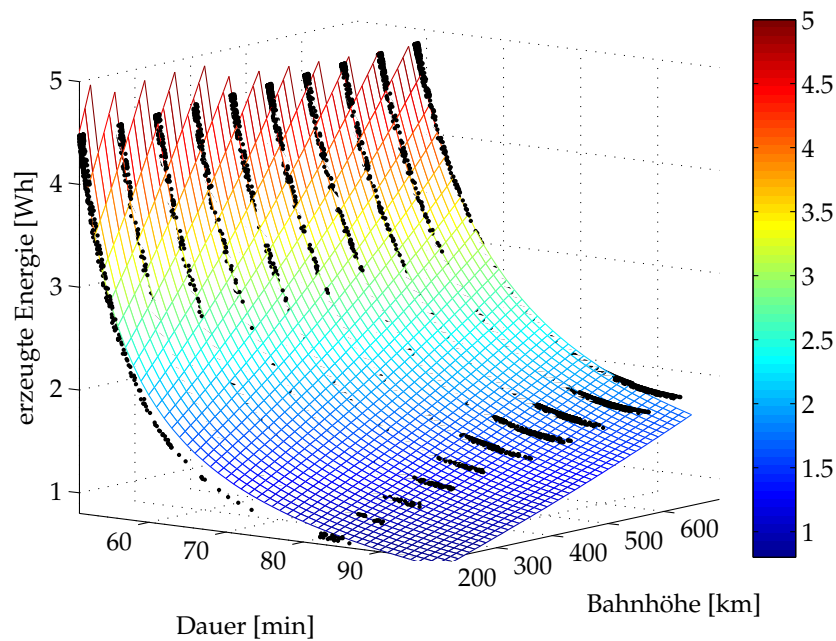
die Argumente der Logarithmen hier dimensionsbehaftet sind. Die Dauer der Sonnenlichtperiode muss daher in Minuten und die Bahnhöhe muss in Kilometern angegeben werden. Die Energie ergibt sich dann in Wattstunden. Der maximale Fehler tritt im Bereich kurzer Sonnenlichtdauern auf und beträgt ca. 12,8 %. Der durchschnittliche Fehler liegt bei ungefähr 1,9 %.

$q_1 =$	3,5412	[Wh]	$q_5 =$	$- 2,8749 \cdot 10^{-1}$	[-]
$q_2 =$	$- 7,9658 \cdot 10^{-2}$	[Wh/min]	$q_6 =$	$5,0701 \cdot 10^{-2}$	[-]
$q_3 =$	$3,4854 \cdot 10^{-3}$	[Wh/km]	$q_7 =$	$- 8,1159 \cdot 10^{-3}$	[-]
$q_4 =$	$- 3,8153 \cdot 10^{-5}$	[Wh/(min · km)]			

**Tabelle 10.7:** Faktoren der Näherungsgleichung (10.8) zur Abschätzung der von den seitlichen Paneelen der Standardkonfiguration erzeugten Energie als Funktion von Sonnenlichtdauer in Minuten und Bahnhöhe in Kilometern

Entsprechend Abbildung 10.7 zeigt Abbildung 10.8 die Verläufe der erzeugten Energie des oberen Paneels als Funktion von Sonnenlichtdauer und Bahnhöhe. Wie erwartet zeigt sich hier der umgekehrte Verlauf: Die durch das obere Paneel erzeugte Energie fällt mit steigender Dauer der Sonnenlichtperiode stark ab. Dabei fällt auf, dass die erzeugte Energie bei gleichbleibender Sonnenlichtdauer mit der Bahnhöhe steigt. Betrachtet man zwei Orbits mit dem gleichen, betragsmäßig kleinen Betawinkel aber verschiedener Bahnhöhe, so ist die Umlaufzeit und damit die Sonnenlichtdauer auf dem niedrigeren Orbit geringer. Möchte man die gleiche Sonnenlichtdauer auf beiden

Orbits erreichen, muss man folglich den Betawinkel des höheren Orbits verringern oder den des niedrigeren Orbits erhöhen. Dies hat zur Folge, dass das obere Paneel auf dem höheren Orbit günstiger zur Sonne steht und damit mehr Energie erzeugt. Für die seitlichen Paneele der Standardkonfiguration ist gerade das Gegenteil richtig: Die erzeugte Energie fällt mit zunehmender Bahnhöhe. Addiert man die Verläufe der seitlichen und des oberen Paneels, erhält man den in Abbildung 10.6 dargestellten näherungsweise linearen Verlauf der Gesamtenergie.



**Abbildung 10.8:** Verlauf der erzeugten Energie des oberen Paneels als Funktion der Dauer der Sonnenlichtperiode und der Bahnhöhe, Simulationsergebnisse als Punkte und Verlauf nach der Näherungsgleichung (10.8), siehe auch Abbildung G.4

Die Faktoren der dabei verwendeten Näherungsgleichung sind in Tabelle 10.8 gegeben, wobei die gleichen Einschränkungen bezüglich der verwendeten Einheiten wie bei der durch die seitlichen Paneele erzeugten Energie berücksichtigt werden müssen. Der maximale Fehler tritt hier auf niedrigen Bahnen bei mittleren Sonnenlichtdauern von ca. 70 Minuten auf und beträgt ca. 15,5 %. Durchschnittlich ist der Fehler durch die Näherung ca. 1,9 %.

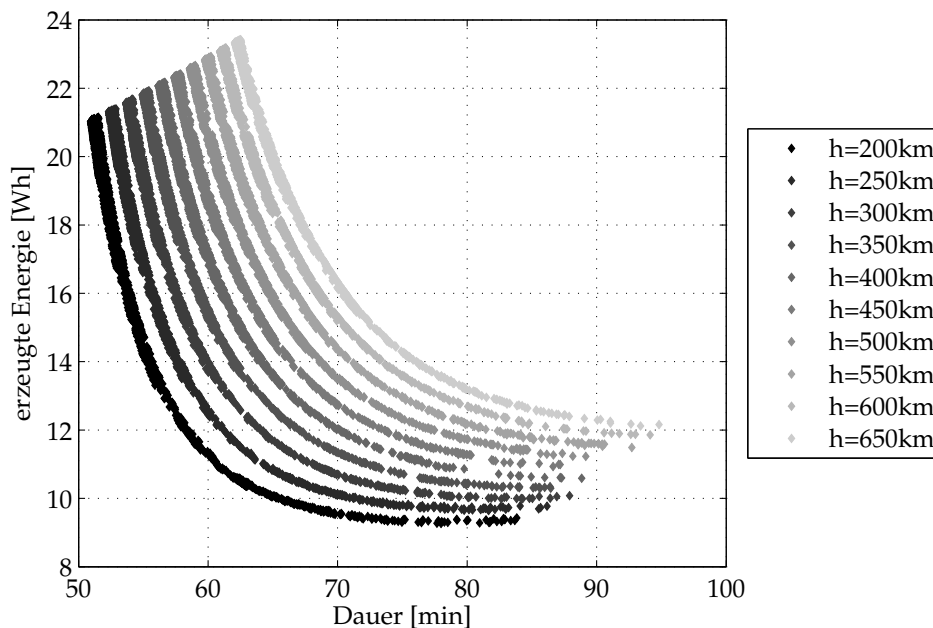
$q_1 =$	$-$	$9,4787 \cdot 10^{-2}$	[Wh]	$q_5 =$	$-$	$2,7974 \cdot 10^{-1}$	[-]
$q_2 =$		$3,5310 \cdot 10^{-4}$	[Wh/min]	$q_6 =$	$-$	$9,1747 \cdot 10^{-2}$	[-]
$q_3 =$		$1,0739 \cdot 10^{-5}$	[Wh/km]	$q_7 =$		$2,7232 \cdot 10^{-2}$	[-]
$q_4 =$	$-$	$9,6471 \cdot 10^{-7}$	[Wh/(min · km)]				

**Tabelle 10.8:** Faktoren der Näherungsgleichung (10.8) zur Abschätzung der vom oberen Paneel erzeugten Energie als Funktion von Sonnenlichtdauer in Minuten und Bahnhöhe in Kilometern



### 10.3.4 Konfigurationen mit ausklappbaren Paneelen

Für die Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen zeigt Abbildung 10.9 analog zu Abbildung 10.6 für die Standardkonfiguration die insgesamt erzeugte Energie als Funktion der Sonnenlichtdauer für verschiedene Bahnhöhen. Die Verläufe für die verschiedenen Bahnhöhen lassen sich dabei im Gegensatz zu der Standardkonfiguration deutlich trennen, wie in der Abbildung zu erkennen ist. Dies liegt daran, dass für die seitlichen Paneel im ausgeklappten Zustand das gleiche gilt wie für das obere Paneel: Bei gleicher Sonnenlichtdauer steigt die erzeugte Energie mit zunehmender Bahnhöhe aufgrund des günstigeren Betawinkels. Ähnlich wie bei dem oberen Paneel in Abbildung 10.8 zu erkennen zeigt sich auch hier eine Abnahme der erzeugten Energie mit zunehmender Dauer der Sonnenlichtperiode, da die seitlichen Paneele ähnlich dem oberen Paneel zur Sonne orientiert sind.

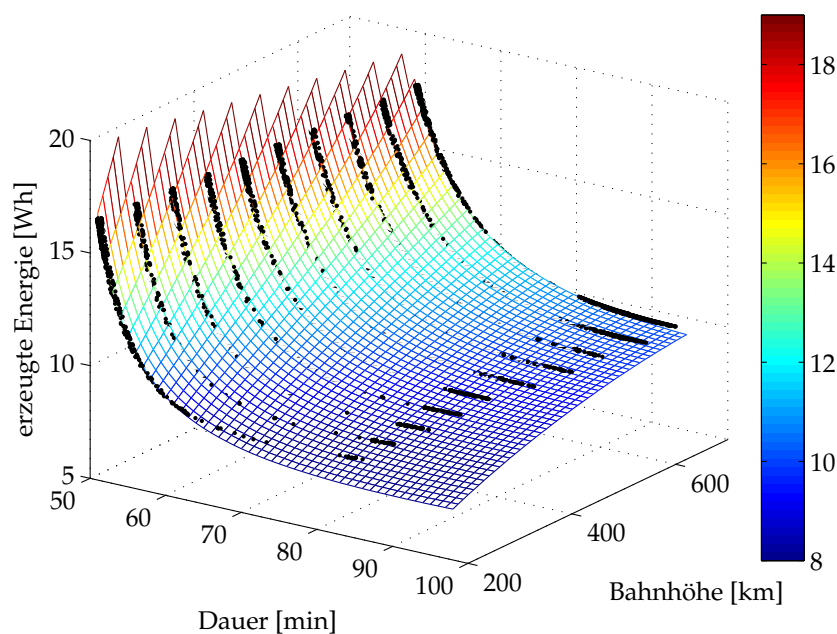


**Abbildung 10.9:** Streudiagramm der von der Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen erzeugten Gesamtenergie als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen

Vergleicht man die absoluten Werte der erzeugten Energie der Standardkonfiguration und der Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen zeigt sich, dass die Standardkonfiguration im besten Fall weniger Energie erzeugt, im schlechtesten Fall jedoch deutlich mehr. Sie ist also insbesondere dann besser geeignet, wenn die Dauer der Sonnenlichtperioden stark schwankt, also auf Orbits mit mittlerer bis hoher Inklination unterhalb von  $90^\circ$ . Darüber hinaus ist die Standardkonfiguration zu bevorzugen, wenn konstant lange Sonnenlichtperioden zu erwarten sind, also insbesondere auf SSOs, deren LTDN im Bereich der Dämmerung liegt. Herrschen hingegen kurze Sonnenlichtzeiten vor, wie auf niedrig inklinierten, insbesondere auf äquatorialen Orbits, sowie auf SSOs mit einer LTDN im Bereich von 0 h oder 12 h, ist

eine Konfiguration mit ausklappbaren Paneelen besser geeignet.

Die von der Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen insgesamt erzeugte Energie ergibt sich als Summe der durch das obere Paneel erzeugten Energie, wie im vorherigen Abschnitt gezeigt und der durch die ausgeklappten seitlichen Paneelen erzeugten Energie nach Gleichung (10.6). Deren Verlauf ist in Abbildung 10.10 als Funktion der Sonnenlichtdauer und der Bahnhöhe gezeigt. Man erkennt hier ebenfalls einen ähnlichen Verlauf wie bei der vom oberen Paneel erzeugten Energie, wobei die Steigung aufgrund der Neigung der Paneele hier insgesamt geringer ausfällt.



**Abbildung 10.10:** Verlauf der erzeugten Energie aller seitlichen Paneele der Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen als Funktion der Dauer der Sonnenlichtperiode und der Bahnhöhe, Simulationsergebnisse als Punkte und Verlauf nach der Näherungsgleichung (10.8), siehe auch Abbildung G.5

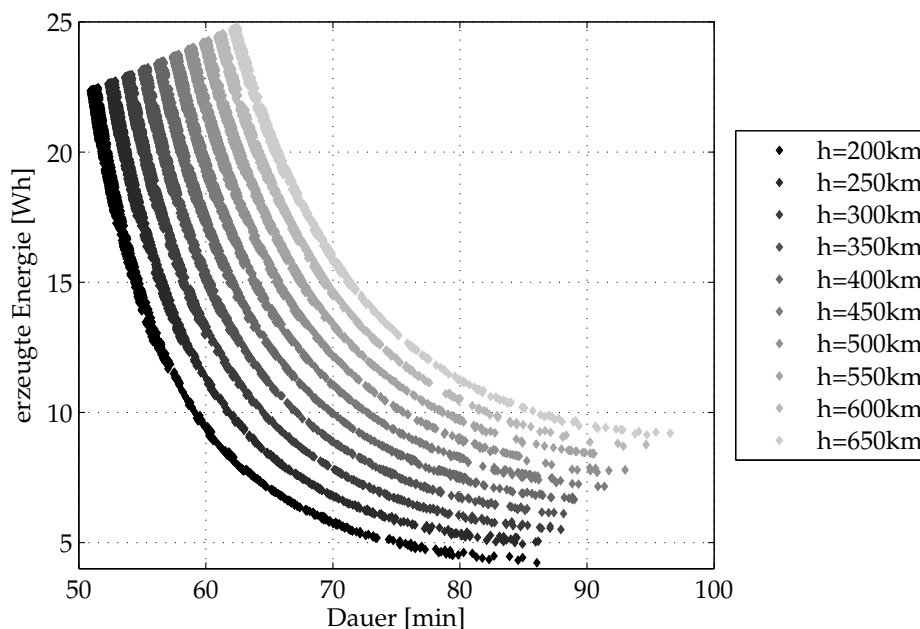
Für die Näherung der simulierten Werte mit einer mathematischen Funktion wird wiederum Gleichung (10.8) herangezogen. Die dabei verwendeten Koeffizienten sind in Tabelle 10.9 zusammengefasst. Dabei tritt der maximale Fehler von ca. 15,2 % auf niedrigen Bahnen bei langen Sonnenlichtperioden auf. Der durchschnittliche Fehler beträgt ca. 1,4 %.

Auf ähnliche Weise kann auch die Konfiguration mit um  $90^\circ$  ausgeklappten Paneelen betrachtet werden. Die erzeugte Energie als Funktion der Sonnenlichtdauer und der Bahnhöhe ergibt sich aus Gleichung (10.7), wobei hier für die vom oberen und von den seitlichen Paneelen erzeugte Energie Gleichung (10.8) mit den Koeffizienten aus Tabelle 10.8 verwendet wird, da alle Paneele die gleiche Ausrichtung zur Sonne besitzen. Abbildung 10.11 zeigt die gesamte von dieser Konfiguration erzeugte Energie als

$q_1 =$	$2,0911 \cdot 10^{-1}$	[Wh]	$q_5 =$	$- 2,6978 \cdot 10^{-1}$	[-]
$q_2 =$	$- 7,8810 \cdot 10^{-3}$	[Wh/min]	$q_6 =$	$- 1,2707 \cdot 10^{-1}$	[-]
$q_3 =$	$- 1,2222 \cdot 10^{-4}$	[Wh/km]	$q_7 =$	$3,4782 \cdot 10^{-2}$	[-]
$q_4 =$	$5,8212 \cdot 10^{-6}$	[Wh/(min · km)]			

**Tabelle 10.9:** Faktoren der Näherungsgleichung (10.8) zur Abschätzung der von den seitlichen Paneelen der Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen erzeugten Energie als Funktion von Sonnenlichtdauer in Minuten und Bahnhöhe in Kilometern

Funktion der Sonnenlichtdauer für verschiedene Bahnhöhen.



**Abbildung 10.11:** Streudiagramm der von der Konfiguration mit um  $90^\circ$  ausgeklappten Paneelen erzeugten Gesamtenergie als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen

Der für die Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen beschriebene Trend zeigt sich hier verstärkt. Mit zunehmender Dauer der Sonnenlichtperiode fällt die erzeugte Energie stark ab, die Differenz zwischen dem besten und dem schlechtesten Fall wird größer. Bei sehr kurzen Sonnenlichtdauern erzeugt diese Konfiguration folglich die meiste Energie. Wie in Abschnitt 9.1.4 bereits ausgeführt wurde, muss dafür jedoch das DEPS entsprechend angepasst werden, da alle Paneele während eines kurzen Zeitraumes sehr viel Leistung zur Verfügung stellen und während eines relativ langen Zeitraumes keine Energie von den Solarpaneelen erzeugt wird. Dies führt während eines Orbits zu einer großen DoD und während der Zeit, in der die Paneele Energie erzeugen, zu einer großen Laderate. Die Batterie muss entsprechend darauf ausgelegt werden.

Mit den in Abschnitt 10.2 beschriebenen Näherungsgleichungen lassen sich für einen beliebigen Orbit mit Hilfe der Bahnhöhe und der Inklination bzw. der LTDN die mittlere, kürzeste und längste Sonnenlichtdauer bestimmen und daraus die dazugehörige erzeugte Energie über die in diesem Abschnitt gegebenen Gleichungen. Dabei treten jedoch, wie im Zusammenhang mit den einzelnen Gleichungen erwähnt, bei bestimmten Werten merkliche Fehler auf. Darüber hinaus lassen sich aus der Betrachtung eines durchschnittlichen und eines schlechtesten Falls der erzeugten Energie über ein Jahr als Funktion des Orbits einige weitere Schlüsse ziehen, weshalb auch diese Ergebnisse im folgenden Abschnitt vorgestellt werden.

## 10.4 Anhaltswerte der erzeugten Energie als Funktion der Bahnelemente

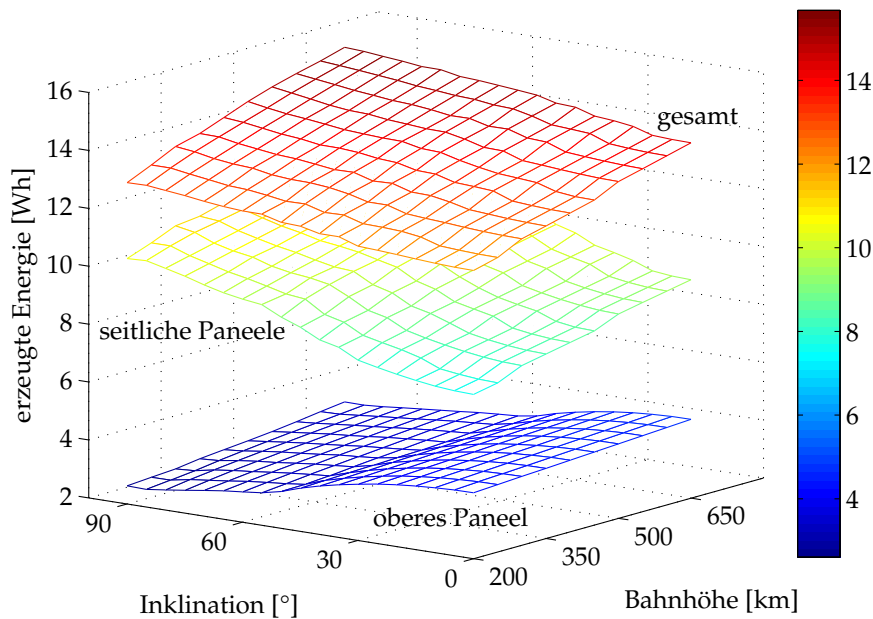
In diesem Abschnitt werden jeweils durchschnittliche und schlechteste Fälle der erzeugten Energie der verschiedenen Konfigurationen über eine Mission von einem Jahr als Funktion der relevanten Bahnelemente dargestellt. Die zur Simulation verwendeten Eingabeparameter wurden in Abschnitt 8.1.2 vorgestellt. Um die über eine Sonnenlichtperiode erzeugte Energie bestimmen zu können, muss dabei eine in der STK-Simulation vorhandene Sonnenlichtperiode genutzt werden. Für den durchschnittlichen Fall wird dabei jeweils die Sonnenlichtperiode betrachtet, deren Dauer dem arithmetischen Mittel der Längen aller Sonnenlichtperioden, die kürzer sind als ein Umlauf, am nächsten kommt. Für den schlechtesten Fall wird bei der Standardkonfiguration die kürzeste Sonnenlichtperiode und für die Konfiguration mit ausklappbaren Paneelen die längste Sonnenlichtperiode, die kürzer ist als die Umlaufzeit, betrachtet.

Dabei werden jeweils die seitlichen Paneele getrennt von dem oberen Paneel untersucht, so dass hier ebenfalls Gleichung (10.6) bzw. Gleichung (10.7) zur Berechnung der Gesamtenergie verwendet werden können. Dabei muss beachtet werden, dass die minimal erzeugte Energie auf die Gesamtkonfiguration bezogen ist. Da das obere Paneel und die seitlichen Paneele auf unterschiedlichen Bahnen unterschiedlich viel Energie erzeugen, kommt es zum Beispiel vor, dass das obere Paneel im Fall der minimalen Energie mehr Energie erzeugt als im Fall der durchschnittlichen Energie.

### 10.4.1 Standardkonfiguration

#### Orbits mit Inklinationen zwischen $0^\circ$ und $90^\circ$

Abbildung 10.12 zeigt die über eine Jahresmission durchschnittlich erzeugte Energie des oberen Paneels, der seitlichen Paneele sowie der Standardkonfiguration insgesamt als Funktion der Bahnhöhe und der Inklination. Dabei wird hier wiederum der Mittelwert aller Szenarien mit verschiedenen Startzeitpunkten und -rektaszensionen in den Diagrammen dargestellt. Auch hier zeigt sich wie schon bei den Sonnenlichtzeiten eine deutliche Abhängigkeit von Bahnhöhe und Inklination, wobei die Abhängigkeit von Startzeitpunkt und -rektaszension vernachlässigbar ist (siehe auch Abschnitt 10.2).



**Abbildung 10.12:** Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.6

Die Verläufe lassen sich mit Hilfe von Abbildung 10.2 erklären. Je höher die Inklination, um so größer ist auch die durchschnittliche Sonnenlichtdauer, was bei dem oberen Paneel zu einer niedrigeren Energieausbeute und bei den seitlichen Paneelen zu einer höheren Energieausbeute führt. Darüber hinaus steigt die Energieausbeute der Paneele wie auch die Sonnenlichtdauer mit der Bahnhöhe. Insgesamt ist also ein möglichst hoher Orbit mit hoher Inklination am günstigsten.

Man erkennt sowohl für das obere Paneel als auch für die seitlichen Paneele und damit auch für die Gesamtkonfiguration näherungsweise lineare Verläufe der Form

$$E(h, i) = q_1 + q_2 \cdot h + q_3 \cdot i \quad (10.9)$$

Die Linearität erklärt sich aus den bisherigen Darstellungen. Wie in Abschnitt 10.2 gezeigt wird, verläuft die durchschnittliche Sonnenlichtdauer ebenfalls näherungsweise linear über die Bahnhöhe und die Inklination. Aus den Diagrammen der erzeugten Energie als Funktion der Bahnhöhe und der Sonnenlichtdauer in Abschnitt 10.3.3 geht des weiteren hervor, dass die Verläufe über den hier relevanten Bereich der Dauer in erster Näherung ebenfalls als linear angesehen werden können. Die für die Näherungsgleichungen für den Verlauf der vom oberen und von den seitlichen Paneelen erzeugten Energie verwendeten Faktoren sind in Tabelle 10.10 zusammengefasst. Bei dem oberen Paneel schwanken die Werte im Bereich von hohen Inklinationen aufgrund des verringerten Einflusses der  $J_2$ -Störung relativ stark, was in diesem Bereich zu einem relativ großen maximalen Fehler von ca. 21,3 % führt, wobei der mittlere Fehler ca. 2,5 % beträgt. Bei den seitlichen Paneelen ist der maximale Fehler ca. 6,0 %, ebenfalls

im Bereich hoher Inklinationen, und der durchschnittliche Fehler 1,4 %.

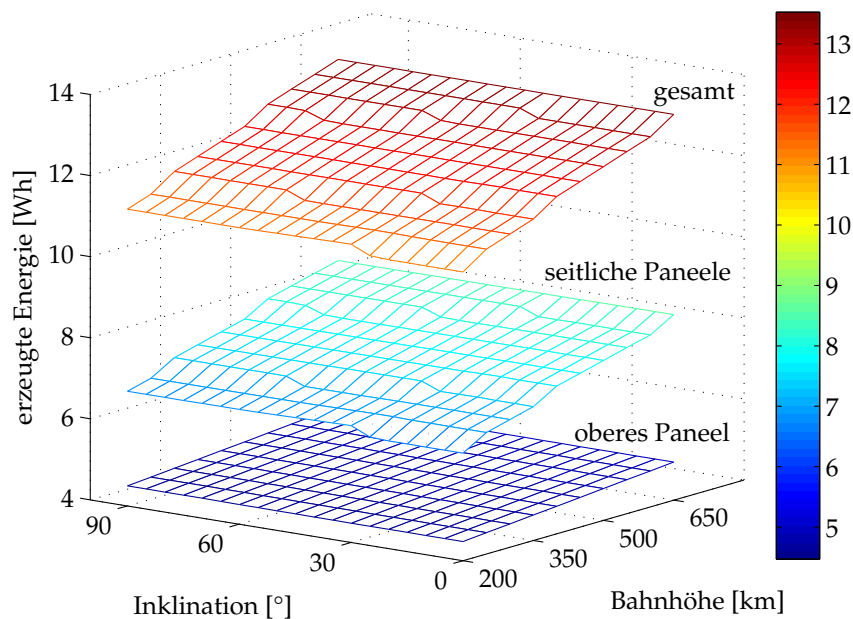
oberes Paneel			seitliche Paneele		
$q_1 =$	4,0923	[min]	$q_1 =$	6,6264	[min]
$q_2 =$	$1,2435 \cdot 10^{-3}$	[min/km]	$q_2 =$	$4,3603 \cdot 10^{-3}$	[min/km]
$q_3 =$	$-1,9113 \cdot 10^{-2}$	[min/°]	$q_3 =$	$3,5130 \cdot 10^{-2}$	[min/°]

**Tabelle 10.10:** Faktoren der Näherungsgleichung (10.9) zur Abschätzung der von den seitlichen Paneelen und des oberen Paneels der Standardkonfiguration durchschnittlich erzeugten Energie als Funktion der Bahnhöhe und der Inklination

Wie bereits erwähnt lässt sich mit Hilfe der Näherungsgleichung auch der Einfluss fehlender Paneele abschätzen, wenn zum Beispiel größere Nutzlasten untergebracht werden müssen. Fehlte zum Beispiel das obere Paneel, würde im Mittel zwischen 19 % und 37 % weniger Energie erzeugt, wobei der Verlust bei niedrigen Inklinationen größer ist. Pro fehlendem seitlichen Paneel würde zwischen 15 % und 20 % weniger Energie zur Verfügung stehen, wobei dieser Verlust insbesondere auf Orbits großer Inklination zum Tragen kommt.

Abbildung 10.13 zeigt analog zu Abbildung 10.12 die von den seitlichen und dem oberen Paneel bei minimaler Gesamtenergie erzeugte Energie als Funktion der Bahnhöhe und der Inklination. Grundsätzlich zeigt sich bei den seitlichen Paneelen und damit auch bei der Gesamtenergie ein ähnlicher Verlauf wie bei der durchschnittlich erzeugten Energie; die erzeugte Gesamtenergie steigt mit der Bahnhöhe und der Inklination. Wie in Abschnitt 3.2 ausgeführt, ist die maximale Eklipsendauer und damit die minimale Sonnenlichtdauer unter Annahme einer kugelförmigen Erde ausschließlich eine Funktion der Bahnhöhe. Unter realistischeren Annahmen kommt auch eine geringe Abhängigkeit von der Inklination hinzu (siehe auch Abschnitt 10.2), die sich auch in den Verläufen der erzeugten Energie zeigt. Dieser Effekt ist bei den seitlichen Paneelen stärker als bei dem oberen Paneel, da insbesondere die seitlichen Paneele zu Beginn und am Ende der Sonnenlichtphase beschienen werden.

Auch hier kann die lineare Gleichung (10.9) zur Annäherung der Werte herangezogen werden. Die dabei verwendeten Koeffizienten sind in Tabelle 10.11 zusammengefasst. Man erkennt den geringeren Einfluss der Inklination im Vergleich zu den Werten für die durchschnittlich erzeugte Energie anhand der Größenordnung der Koeffizienten  $q_3$ . Bei der vom oberen Paneel durchschnittlich erzeugten Energie war dieser Koeffizient aufgrund der längeren Sonnenlichtdauern bei höheren Inklinationen und der damit verbundenen Reduktion der erzeugten Energie negativ. Bei der minimal erzeugten Energie wird er aufgrund des Einflusses der Abplattung der Erde positiv. Für den Verlauf der Werte für das obere Paneel gilt die Annahme einer linearen Funktion sehr genau, der maximal bei der Näherung auftretende Fehler beträgt lediglich 0,2 %. Bei den seitlichen Paneele ist der maximale Fehler ca. 2,7 % und der durchschnittliche Fehler ca. 0,6 %.



**Abbildung 10.13:** Minimalwerte der durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugten Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.7

oberes Paneel		seitliche Paneele	
$q_1 = 4,2606$	[min]	$q_1 = 5,9754$	[min]
$q_2 = 1,0411 \cdot 10^{-3}$	[min/km]	$q_2 = 3,8944 \cdot 10^{-3}$	[min/km]
$q_3 = 2,6694 \cdot 10^{-4}$	[min/°]	$q_3 = 1,2984 \cdot 10^{-3}$	[min/°]

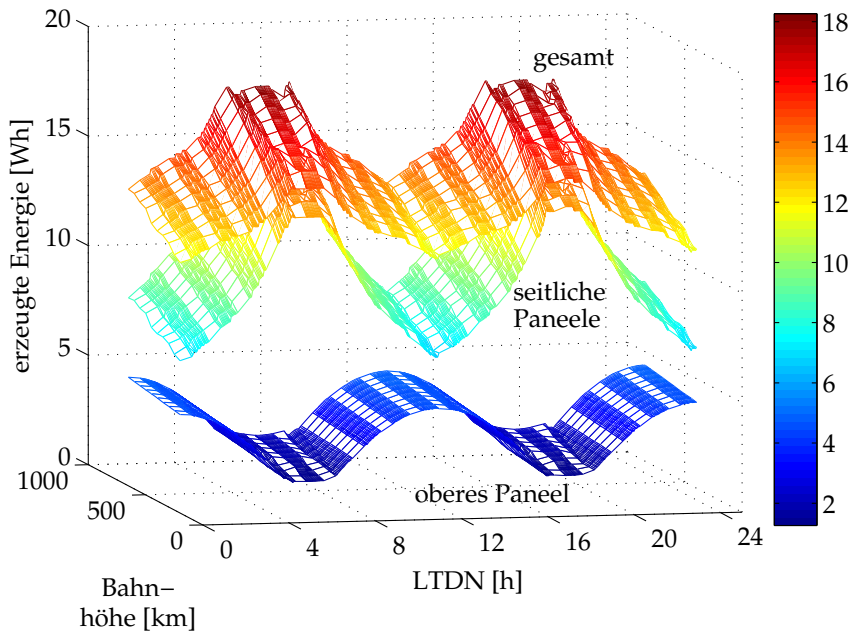
**Tabelle 10.11:** Faktoren der Näherungsgleichung (10.9) zur Abschätzung der von den seitlichen Paneelen und des oberen Paneels der Standardkonfiguration minimal erzeugten Energie als Funktion der Bahnhöhe und der Inklination

Während der hier betrachteten kürzesten Sonnenlichtzeiten trägt das obere Paneel am meisten zur erzeugten Energie bei. Wäre es nicht vorhanden, würde daher ca. 37 % weniger Energie zur Verfügung stehen, während ein fehlendes seitliches Paneel nur ca. 15 % Energieverlust zur Folge hätte.

### Sonnensynchrone Orbits

Abbildung 10.14 zeigt die auf SSOs von der Standardkonfiguration durchschnittlich erzeugte Energie als Funktion der Bahnhöhe und der LTDN, wobei wiederum die Verläufe für das obere Paneel, die seitlichen Paneele und die Gesamtkonfiguration dargestellt sind. Die Abhängigkeit von der Bahnhöhe ist ähnlich wie auf Orbits niedrigerer Inklination, die Abhängigkeit von der LTDN zeigt den erwarteten Verlauf: Das obere Paneel erzeugt auf Dämmerungsorts, auf denen lange Sonnenlichtzeiten auftreten, relativ wenig Energie und auf Orbits mit einer LTDN im Bereich von 0 h oder 12 h

relativ viel Energie. Für die seitlichen Paneele ist erwartungsgemäß genau das Gegenteil der Fall. Insgesamt betrachtet heben sich die Effekte ein Stück weit auf, wobei der Einfluss der seitlichen Paneele größer ist, weshalb der Verlauf der durch die Gesamtkonfiguration erzeugten Energie dem der durch die seitlichen Paneele erzeugten Energie ähnelt. Insgesamt kann auf günstigen SSOs im Durchschnitt mehr Energie erzeugt werden, als auf niedriger inklinierten Orbits, da hier die Schwankungen in den Sonnenlichtzeiten deutlich geringer sind.



**Abbildung 10.14:** Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der LTDN, siehe auch Abbildung G.8

Ähnlich wie bei den Sonnenlichtzeiten auf den SSOs (siehe Abschnitt 10.2.3) werden auch hier die entsprechenden Näherungsgleichungen nur für den Bereich der LTDN zwischen 0h und 12h angegeben, wobei auch hier eine abschnittsweise definierte Funktion verwendet wird. Dabei bietet sich hier ein kubisches Polynom zweier Variablen, nämlich der Bahnhöhe und der LTDN an. Es ist

$$E(h, LTDN) = \begin{cases} q_1 + q_2 \cdot h + q_3 \cdot LTDN + q_4 \cdot h^2 + q_5 \cdot LTDN^2 + q_6 \cdot h^3 + \dots \\ \dots + q_7 \cdot LTDN^3 + q_8 \cdot h \cdot LTDN + q_9 \cdot h^2 \cdot LTDN + q_{10} \cdot h \cdot LTDN^2, & 0h \leq LTDN < 6h \\ q_{11} + q_{12} \cdot h + q_{13} \cdot LTDN + q_{14} \cdot h^2 + q_{15} \cdot LTDN^2 + q_{16} \cdot h^3 + \dots \\ \dots + q_{17} \cdot LTDN^3 + q_{18} \cdot h \cdot LTDN + q_{19} \cdot h^2 \cdot LTDN + q_{20} \cdot h \cdot LTDN^2, & 6h \leq LTDN \leq 12h \end{cases} \quad (10.10)$$

Die dargestellten Verläufe verflachen im Bereich einer LTDN von 6h aufgrund der Tatsache, dass hier nur Sonnenlichtperioden berücksichtigt werden, die kürzer sind als die Umlaufdauer. Daher gibt es bei den im Folgenden vorgestellten Verläufen jeweils



eine minimale bzw. maximale Grenzenergie als lineare Funktion der Bahnhöhe, die die Ober- bzw. Untergrenze der erzeugten Energie darstellt. Ist der Funktionswert der angegebenen Näherungsgleichungen größer bzw. kleiner als dieser Wert, wird die Funktion auf diesen Wert gesetzt.

Für die seitlichen Paneele ist diese Grenzenergie durch die folgende Gleichung gegeben, die direkt aus Simulationswerten der Energie bei einer LTDN von 6 h bei 200 km und 650 km Bahnhöhe bestimmt wurde.

$$E_{\text{grenz,ave,side}}^{\text{max}}(h) = 5,6889 \cdot 10^{-3} \frac{\text{Wh}}{\text{km}} \cdot h + 12,222 \text{ Wh} \quad (10.11)$$

Die für die Näherung der durch die seitlichen Paneele erzeugte Energie verwendeten Koeffizienten sind in Tabelle 10.12 zusammengefasst. Dabei muss beachtet werden, dass die hier vorkommenden Potenzen der Dauer und der Bahnhöhe zu relativ großen Werten führen, was dazu führt, dass die numerischen Werte einiger Koeffizienten wie zum Beispiel  $q_6$  und  $q_{16}$  sehr klein werden. Als Vorfaktor vor den entsprechend großen Potenzen der Variablen können sie dennoch nicht vernachlässigt werden. Der maximale Fehler tritt dabei in der Nähe der LTDN von 6 h auf und beträgt ca. 4,8 %. Der durchschnittliche Fehler beträgt ca. 1 %. Abbildung 10.15 zeigt die Simulationsergebnisse als Punkte und die Näherung als farbigen Verlauf.

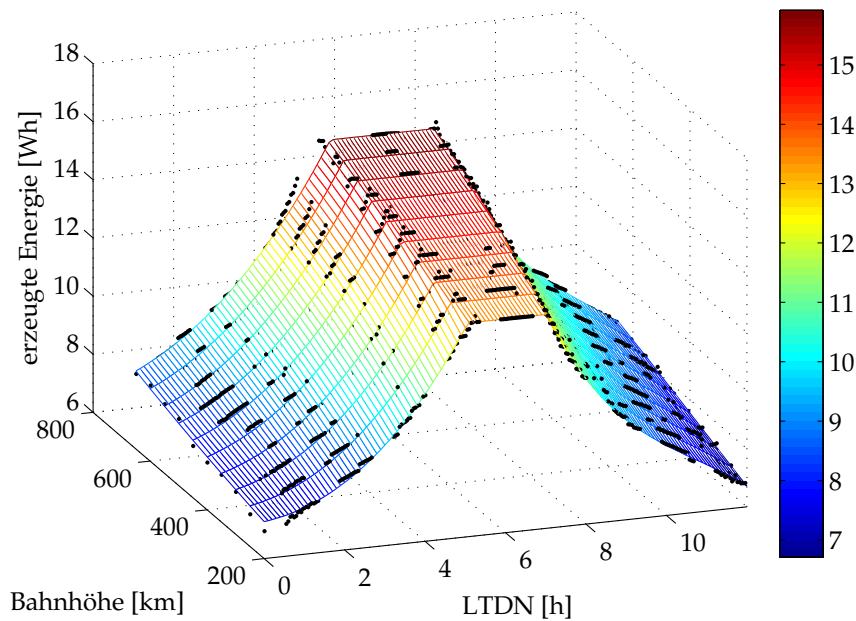
$q_1 =$	6,8192	[Wh]	$q_{11} =$	$7,1006 \cdot 10^1$	[Wh]
$q_2 =$	$2,0788 \cdot 10^{-3}$	[Wh/km]	$q_{12} =$	$4,3680 \cdot 10^{-2}$	[Wh/km]
$q_3 =$	$- 2,7749 \cdot 10^{-1}$	[Wh/h]	$q_{13} =$	$- 1,7256 \cdot 10^1$	[Wh/h]
$q_4 =$	$9,0526 \cdot 10^{-7}$	[Wh/km <sup>2</sup> ]	$q_{14} =$	$- 9,4445 \cdot 10^{-6}$	[Wh/km <sup>2</sup> ]
$q_5 =$	$1,7840 \cdot 10^{-1}$	[Wh/h <sup>2</sup> ]	$q_{15} =$	1,5990	[Wh/h <sup>2</sup> ]
$q_6 =$	$4,6800 \cdot 10^{-10}$	[Wh/km <sup>3</sup> ]	$q_{16} =$	$- 1,7983 \cdot 10^{-9}$	[Wh/km <sup>3</sup> ]
$q_7 =$	$8,0161 \cdot 10^{-3}$	[Wh/h <sup>3</sup> ]	$q_{17} =$	$- 5,1067 \cdot 10^{-2}$	[Wh/h <sup>3</sup> ]
$q_8 =$	$2,5354 \cdot 10^{-3}$	[Wh/(km·h)]	$q_{18} =$	$- 6,0542 \cdot 10^{-3}$	[Wh/(km·h)]
$q_9 =$	$- 1,6627 \cdot 10^{-6}$	[Wh/(km <sup>2</sup> ·h)]	$q_{19} =$	$9,5840 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]
$q_{10} =$	$- 6,9987 \cdot 10^{-5}$	[Wh/(km·h <sup>2</sup> )]	$q_{20} =$	$2,2524 \cdot 10^{-4}$	[Wh/(km·h <sup>2</sup> )]

**Tabelle 10.12:** Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch die seitlichen Paneele der Standardkonfiguration durchschnittlich erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien unter  $E_{\text{grenz,ave,side}}^{\text{max}}$

Die für die Näherung der durch das obere Paneel bei durchschnittlicher Gesamtenergie erzeugten Energie verwendeten Koeffizienten sind in Tabelle 10.13 gegeben. Dabei muss beachtet werden, dass die Energie eine minimale Grenzenergie nicht unterschreitet, die gegeben ist durch

$$E_{\text{grenz,ave,top}}^{\text{min}}(h) = 1,9222 \cdot 10^{-3} \frac{\text{Wh}}{\text{km}} \cdot h + 0,9126 \text{ Wh} \quad (10.12)$$

Der bei dieser Näherung auftretende maximale Fehler beträgt ca. 11,4 %, der durchschnittliche Fehler ca. 1,9 %.

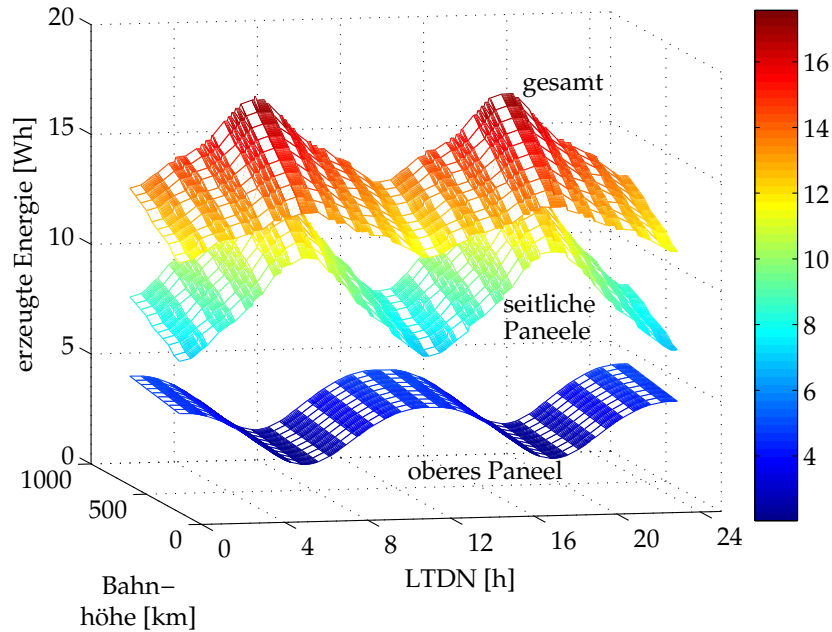


**Abbildung 10.15:** Durchschnittlich durch die seitlichen Paneele der Standardkonfiguration erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, Simulationsergebnisse als Punkte und angenäherter Verlauf, siehe auch Abbildung G.8

$q_1 =$	4,7648	[Wh]	$q_{11} =$	—	1,6642	[Wh]	
$q_2 =$	—	$4,8504 \cdot 10^{-3}$	[Wh/km]	$q_{12} =$	$1,2994 \cdot 10^{-2}$	[Wh/km]	
$q_3 =$		$5,4804 \cdot 10^{-1}$	[Wh/h]	$q_{13} =$	—	1,3717	[Wh/h]
$q_4 =$		$1,5147 \cdot 10^{-5}$	[Wh/km <sup>2</sup> ]	$q_{14} =$	—	$1,5163 \cdot 10^{-5}$	[Wh/km <sup>2</sup> ]
$q_5 =$	—	$3,7239 \cdot 10^{-1}$	[Wh/h <sup>2</sup> ]	$q_{15} =$		$3,5870 \cdot 10^{-1}$	[Wh/h <sup>2</sup> ]
$q_6 =$	—	$1,1037 \cdot 10^{-8}$	[Wh/km <sup>3</sup> ]	$q_{16} =$		$1,0007 \cdot 10^{-8}$	[Wh/km <sup>3</sup> ]
$q_7 =$		$2,8058 \cdot 10^{-2}$	[Wh/h <sup>3</sup> ]	$q_{17} =$	—	$1,7248 \cdot 10^{-2}$	[Wh/h <sup>3</sup> ]
$q_8 =$	—	$6,0048 \cdot 10^{-4}$	[Wh/(km·h)]	$q_{18} =$	—	$1,3571 \cdot 10^{-3}$	[Wh/(km·h)]
$q_9 =$	—	$2,1467 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]	$q_{19} =$		$2,8188 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]
$q_{10} =$		$1,5278 \cdot 10^{-4}$	[Wh/(km·h <sup>2</sup> )]	$q_{20} =$		$6,0504 \cdot 10^{-5}$	[Wh/(km·h <sup>2</sup> )]

**Tabelle 10.13:** Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch das obere Paneel durchschnittlich erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien über  $E_{\text{grenz,ave,top}}^{\text{min}}$

Abbildung 10.16 zeigt analog zu Abbildung 10.14 die von dem oberen Paneel, den seitlichen Paneelen und der Gesamtkonfiguration erzeugte Energie bei minimaler Gesamtenergie als Funktion der Bahnhöhe und der LTDN. Grundsätzlich zeigen sich sehr ähnliche Verläufe wie bei der durchschnittlich erzeugten Energie, da die Dauer der Sonnenlichtzeiten auf SSOs wie bereits erwähnt relativ wenig schwankt. Insgesamt ergibt sich insbesondere aufgrund der Unterschiede bei den seitlichen Paneelen jedoch eine etwas niedrigere Gesamtenergie.



**Abbildung 10.16:** Minimale durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.9

Als Näherungsgleichung für die Verläufe wird wieder Gleichung (10.10) verwendet. Für die seitlichen Paneele gibt es wie bei der durchschnittlichen Energie im Bereich einer LTDN von 6 h ebenfalls eine Grenzenenergie für jede Bahnhöhe, die das Maximum der erzeugten Energie darstellt. Sie ergibt sich zu

$$E_{\text{grenz,min,side}}^{\text{max}}(h) = 7,9556 \cdot 10^{-3} \frac{\text{Wh}}{\text{km}} \cdot h + 9,8389 \text{ Wh} \quad (10.13)$$

Die in der Näherungsgleichung für die von den seitlichen Paneelen erzeugte Energie verwendeten Koeffizienten finden sich in Tabelle 10.14. Der maximale dabei auftretende Fehler beträgt ca. 4,4 %, der durchschnittliche Fehler ca. 0,7 %.

Für das obere Paneel existiert ebenfalls eine Grenzenenergie  $E_{\text{grenz,min,top}}^{\text{min}}$ , die bei einer gegebenen Bahnhöhe nicht unterschritten wird. Dabei deutet der tiefgestellte Index *min* an, dass es sich um den Wert für die minimale insgesamt erzeugte Energie handelt und der hochgestellte Index *min*, dass es sich um einen Minimalwert handelt, der nicht unterschritten wird. Für diese Größe gilt:

$$E_{\text{grenz,min,top}}^{\text{min}}(h) = 7,6889 \cdot 10^{-4} \frac{\text{Wh}}{\text{km}} \cdot h + 1,8772 \text{ Wh} \quad (10.14)$$

Auch zur Näherung der durch das obere Paneel erzeugten Energie bei minimaler insgesamt erzeugter Energie wird Gleichung (10.10) herangezogen. Die dabei zu verwendenden Koeffizienten sind in Tabelle 10.15 zusammengetragen. Der maximale Fehler tritt dabei wiederum im Bereich der LTDN von 6 h auf und beträgt ca. 5,1 %, während der durchschnittliche Fehler lediglich ca. 0,5 % beträgt.

$q_1 =$	5,5742	[Wh]	$q_{11} =$	$3,1480 \cdot 10^1$	[Wh]
$q_2 =$	$5,3310 \cdot 10^{-3}$	[Wh/km]	$q_{12} =$	$2,1984 \cdot 10^{-2}$	[Wh/km]
$q_3 =$	$8,2741 \cdot 10^{-1}$	[Wh/h]	$q_{13} =$	– 5,8308	[Wh/h]
$q_4 =$	– $1,8917 \cdot 10^{-6}$	[Wh/km <sup>2</sup> ]	$q_{14} =$	$1,6770 \cdot 10^{-5}$	[Wh/km <sup>2</sup> ]
$q_5 =$	– $6,7359 \cdot 10^{-2}$	[Wh/h <sup>2</sup> ]	$q_{15} =$	$5,3143 \cdot 10^{-1}$	[Wh/h <sup>2</sup> ]
$q_6 =$	$4,7047 \cdot 10^{-10}$	[Wh/km <sup>3</sup> ]	$q_{16} =$	– $1,7315 \cdot 10^{-8}$	[Wh/km <sup>3</sup> ]
$q_7 =$	$1,1156 \cdot 10^{-2}$	[Wh/h <sup>3</sup> ]	$q_{17} =$	– $1,8099 \cdot 10^{-2}$	[Wh/h <sup>3</sup> ]
$q_8 =$	– $8,5670 \cdot 10^{-5}$	[Wh/(km·h)]	$q_{18} =$	– $4,2778 \cdot 10^{-3}$	[Wh/(km·h)]
$q_9 =$	– $1,1229 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]	$q_{19} =$	$3,8616 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]
$q_{10} =$	$1,5433 \cdot 10^{-4}$	[Wh/(km·h <sup>2</sup> )]	$q_{20} =$	$1,7520 \cdot 10^{-4}$	[Wh/(km·h <sup>2</sup> )]

**Tabelle 10.14:** Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch die seitlichen Paneele der Standardkonfiguration minimal erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien unter  $E_{\text{grenz,min,side}}^{\text{max}}$

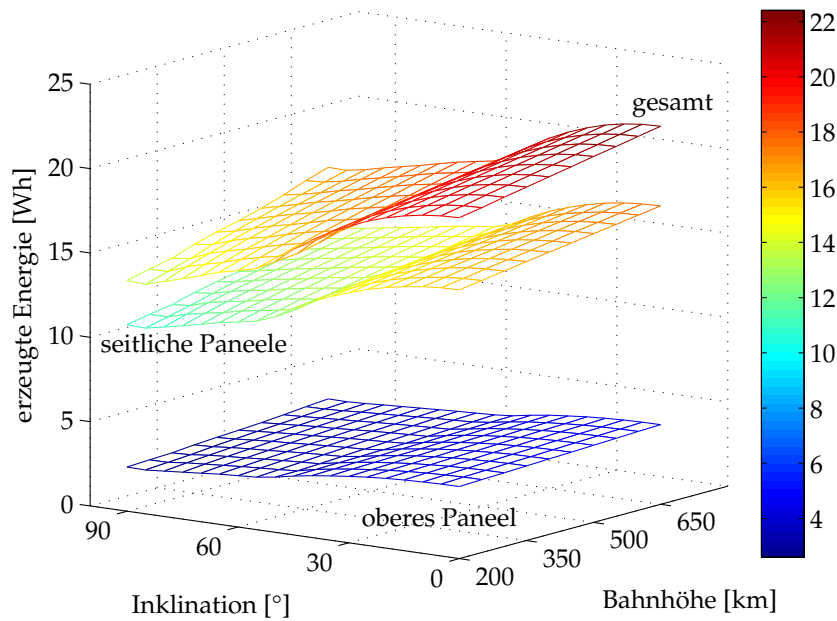
$q_1 =$	4,6400	[Wh]	$q_{11} =$	5,2900	[Wh]
$q_2 =$	– $1,9774 \cdot 10^{-3}$	[Wh/km]	$q_{12} =$	– $2,2571 \cdot 10^{-3}$	[Wh/km]
$q_3 =$	$1,3513 \cdot 10^{-1}$	[Wh/h]	$q_{13} =$	– 2,2371	[Wh/h]
$q_4 =$	$6,9810 \cdot 10^{-6}$	[Wh/km <sup>2</sup> ]	$q_{14} =$	$4,1245 \cdot 10^{-6}$	[Wh/km <sup>2</sup> ]
$q_5 =$	– $1,9943 \cdot 10^{-1}$	[Wh/h <sup>2</sup> ]	$q_{15} =$	$3,6588 \cdot 10^{-1}$	[Wh/h <sup>2</sup> ]
$q_6 =$	– $5,0125 \cdot 10^{-9}$	[Wh/km <sup>3</sup> ]	$q_{16} =$	– $3,0401 \cdot 10^{-9}$	[Wh/km <sup>3</sup> ]
$q_7 =$	$1,6619 \cdot 10^{-2}$	[Wh/h <sup>3</sup> ]	$q_{17} =$	– $1,5442 \cdot 10^{-2}$	[Wh/h <sup>3</sup> ]
$q_8 =$	$1,2665 \cdot 10^{-4}$	[Wh/(km·h)]	$q_{18} =$	$2,7180 \cdot 10^{-4}$	[Wh/(km·h)]
$q_9 =$	– $1,5070 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]	$q_{19} =$	– $1,8419 \cdot 10^{-8}$	[Wh/(km <sup>2</sup> ·h)]
$q_{10} =$	– $1,1513 \cdot 10^{-5}$	[Wh/(km·h <sup>2</sup> )]	$q_{20} =$	– $1,0499 \cdot 10^{-5}$	[Wh/(km·h <sup>2</sup> )]

**Tabelle 10.15:** Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch das obere Paneel der Standardkonfiguration bei minimaler Gesamtenergie erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien unter  $E_{\text{grenz,min,top}}^{\text{min}}$

## 10.4.2 Konfigurationen mit ausklappbaren Paneelen

### Orbits mit Inklinationen zwischen 0° und 90°

Abbildung 10.17 zeigt analog zu Abbildung 10.12 die durchschnittliche von der Konfiguration mit um 60° ausgeklappten Paneelen erzeugte Energie als Funktion der Bahnhöhe und der Inklination. Der Verlauf der vom oberen Paneel erzeugten Energie entspricht dabei exakt dem in Abbildung 10.12 und kann mit Gleichung (10.9) und den in Tabelle 10.10 gegebenen Koeffizienten angenähert werden. Der Verlauf der durch die seitlichen Paneele erzeugten Energie entspricht grob dem des oberen Paneels aufgrund der ähnlichen Ausrichtung zur Sonne, wobei der Einfluss der Inklination deutlich stärker zum Tragen kommt. Dieser ist hier darauf zurückzuführen, dass auf Orbits niedrigerer Inklination kürzere Sonnenlichtperioden vorherrschen. Die hier betrachtete Konfiguration erzeugt also auf Orbits großer Bahnhöhe und geringer Inklination am meisten Energie.



**Abbildung 10.17:** Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.10

Der Verlauf der durch die seitlichen Paneele erzeugten Energie lässt sich mit einem kubischen Polynom zweier Variablen der Form

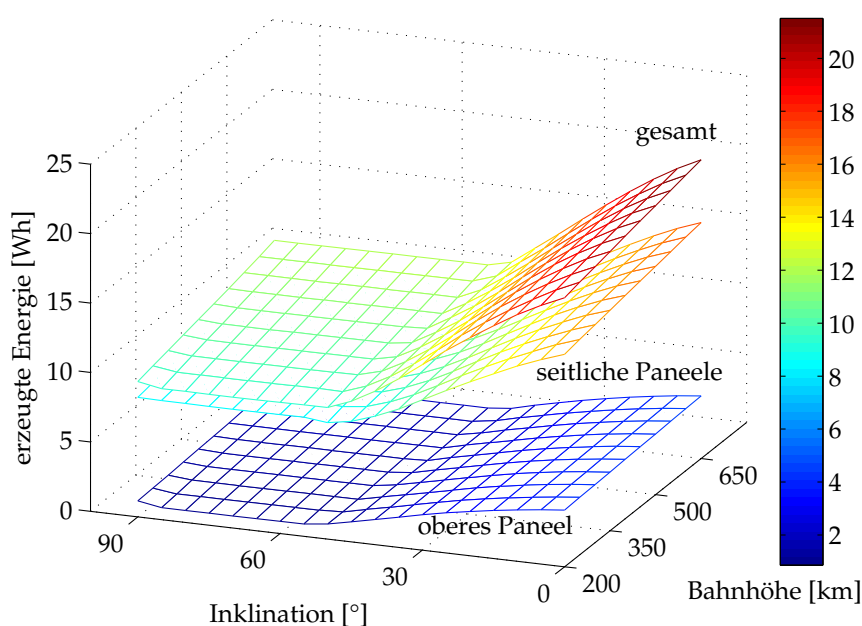
$$E(h, i) = q_1 + q_2 \cdot h + q_3 \cdot i + q_4 \cdot h^2 + q_5 \cdot i^2 + q_6 \cdot h^3 + q_7 \cdot i^3 + q_8 \cdot h \cdot i + q_9 \cdot h^2 \cdot i + q_{10} \cdot h \cdot i^2 \quad (10.15)$$

annähern. Die dabei zu verwendenden Koeffizienten sind in Tabelle 10.16 zusammengefasst. Auch hier tritt der maximale Fehler bei hohen Inklinationen aufgrund der Streuung der Werte auf und beträgt ca. 9,2 %. Der durchschnittliche Fehler ist ca. 1,3 %. Auf den Orbits durchschnittlicher Sonnenlichtdauer wird bei der Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen von jedem Paneel ungefähr gleich viel Energie erzeugt; jedes fehlende Paneel bewirkt also einen Verlust von ca. 20 % der erzeugten Energie.

$q_1 =$	$1,5254 \cdot 10^1$	[Wh]	$q_6 =$	$1,4589 \cdot 10^{-9}$	[Wh/km <sup>3</sup> ]
$q_2 =$	$4,4257 \cdot 10^{-3}$	[Wh/km]	$q_7 =$	$1,2156 \cdot 10^{-5}$	[Wh/° <sup>3</sup> ]
$q_3 =$	$-1,0481 \cdot 10^{-2}$	[Wh/°]	$q_8 =$	$3,4487 \cdot 10^{-6}$	[Wh/(°·km)]
$q_4 =$	$-1,7816 \cdot 10^{-6}$	[Wh/km <sup>2</sup> ]	$q_9 =$	$-1,6070 \cdot 10^{-8}$	[Wh/(°·km <sup>2</sup> )]
$q_5 =$	$-1,6970 \cdot 10^{-3}$	[Wh/° <sup>2</sup> ]	$q_{10} =$	$4,5045 \cdot 10^{-7}$	[Wh/(° <sup>2</sup> ·km)]

**Tabelle 10.16:** Faktoren der Näherungsgleichung (10.15) zur Abschätzung der von den seitlichen Paneelen der Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen durchschnittlich erzeugten Energie als Funktion von Bahnhöhe und Inklination

Für den Fall minimaler insgesamt durch die Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen erzeugter Energie sind die Verläufe der verschiedenen Paneele in Abbildung 10.18 dargestellt. Auch hier zeigen sich grundsätzlich ähnliche Verläufe bei dem oberen Paneel und den seitlichen Paneelen, wobei der Einfluss der Inklination bei den seitlichen Paneelen stärker ist. Man erkennt auf Orbits mit Inklinationen größer als ungefähr  $50^\circ$  einen praktisch konstanten Verlauf der erzeugten Energie bei gegebener Bahnhöhe. Wie bereits erwähnt wird von den Konfigurationen mit ausgeklappten Paneelen besonders wenig Energie erzeugt, wenn die Sonnenlichtdauer sehr lang, aber - für die hier angestellten Betrachtungen - kürzer als die Umlaufdauer ist. Dieser Fall wird auf allen Orbits, deren Inklination größer ist als das in Gleichung (10.2) definierte  $i_{\text{grenz}}$ , erreicht, weshalb sich hier die minimale Energie praktisch nicht ändert. Eine Ausnahme bilden polare Orbits, auf denen wiederum Startzeitpunkt und -rektaszension eine Rolle spielen (siehe auch Abschnitt 10.2.2).



**Abbildung 10.18:** Durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen insgesamt erzeugte Energie bei minimaler Gesamtenergie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.11

Bei der Näherung der Verläufe muss hier beachtet werden, dass die entsprechende Gleichung für die vom oberen Paneel der Standardkonfiguration erzeugte Energie nicht verwendet werden kann, da der Fall der minimal erzeugten Gesamtenergie bei der Standardkonfiguration bei kurzen Sonnenlichtzeiten und bei der Konfiguration mit ausklappbaren Paneelen bei langen Sonnenlichtzeiten auftritt. Daher unterscheiden sich die Verläufe für das obere Paneel für die beiden Fälle. Aufgrund der starken Abflachung des Verlaufes der erzeugten Energie zu hohen Inklinationen hin, bietet sich hier für die Näherung eine Art dreidimensionale gaußsche Glockenkurve mit ei-

nem Offset der Form

$$E(h, i) = q_1 \cdot \exp\left(-\frac{1}{2} \cdot \left(\frac{h - q_2}{q_3}\right)^2\right) + q_4 \cdot \exp\left(-\frac{1}{2} \cdot \left(\frac{i - q_5}{q_6}\right)^2\right) + q_7 \quad (10.16)$$

an. Die Koeffizienten dieser Gleichung zur Annäherung der Verläufe sind in Tabelle 10.17 zusammengefasst. Für die vom oberen Paneel erzeugte Energie ergibt sich dabei ein relativ großer maximaler Fehler von ca. 33 % aufgrund der Abweichungen auf polaren Orbits. Der durchschnittliche Fehler beträgt ca. 6,4 %. Für die seitlichen Paneele tritt der maximale Fehler im Übergangsbereich bei einer Inklination von 45° auf und beträgt ca. 5,1 %, während der durchschnittliche Fehler bei ca. 1,1 % liegt. Diesen Rechnungen zur Folge würde durch das Fehlen des oberen Paneels insbesondere auf niedrig inklinierten Orbits eine relativ starke Reduktion der erzeugten Gesamtenergie von ca. 21 % auftreten, während diese auf hochinklinierten Bahnen im besten Fall nur ca. 10 % beträgt. Dies erklärt sich aus der größeren Steigung der von den seitlichen Paneelen erzeugten Energie mit sinkender Inklination. Das Fehlen eines seitlichen Paneels würde je nach Inklination eine Reduktion zwischen 19 % und 22 % zur Folge haben.

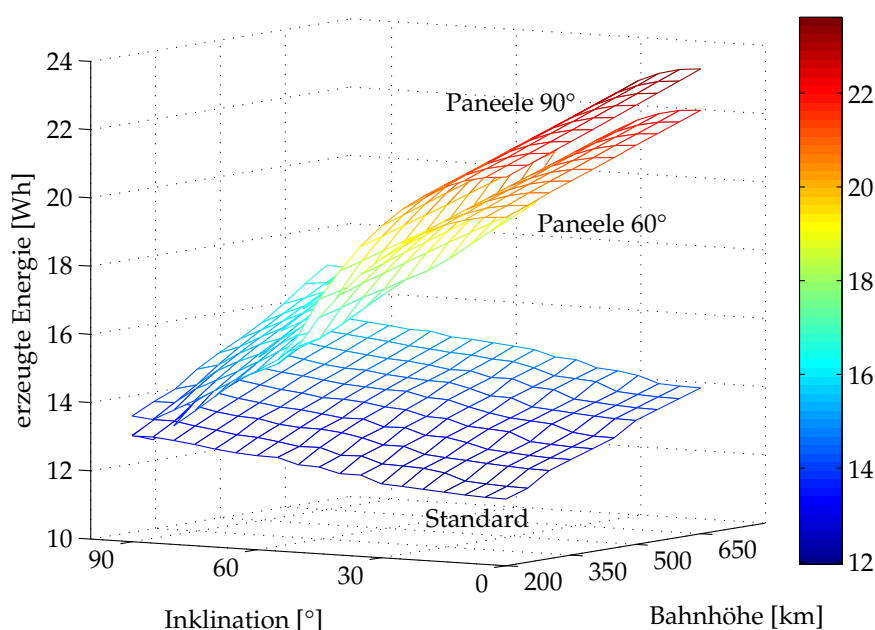
oberes Paneel			seitliche Paneele		
$q_1 =$	$1,2965 \cdot 10^7$	[Wh]	$q_1 =$	$1,5083 \cdot 10^1$	[Wh]
$q_2 =$	$4,0243 \cdot 10^9$	[km]	$q_2 =$	$3,1026 \cdot 10^3$	[km]
$q_3 =$	$1,8120 \cdot 10^9$	[km]	$q_3 =$	$-2,6565 \cdot 10^3$	[km]
$q_4 =$	2,8657	[Wh]	$q_4 =$	6,6709	[Wh]
$q_5 =$	1,6029	[°]	$q_5 =$	-1,0839	[°]
$q_6 =$	$-2,1669 \cdot 10^1$	[°]	$q_6 =$	$1,9961 \cdot 10^1$	[°]
$q_7 =$	$-1,1008 \cdot 10^6$	[Wh]	$q_7 =$	0	[Wh]

**Tabelle 10.17:** Faktoren der Näherungsgleichung (10.16) zur Abschätzung der von den seitlichen Paneelen und des oberen Paneels der Konfiguration mit um 60° ausgeklappten Paneelen bei minimaler Gesamtenergie erzeugten Energie als Funktion der Bahnhöhe und der Inklination

Für die Konfiguration mit um 90° ausgeklappten Paneelen kann die erzeugte Energie mit Hilfe der für die vom oberen Paneel erzeugte Energie bei der Konfiguration mit um 60° ausgeklappten Paneelen angegebenen Näherungsgleichungen für den durchschnittlichen und den minimalen Fall bestimmt werden. Die Verläufe bei der hier getroffenen Annahme, dass alle Paneele mit jeweils 14 Solarzellen bestückt sind, entsprechen denen des oberen Paneels multipliziert mit fünf. Bei dieser Konfiguration sind alle Paneele zu jedem Zeitpunkt gleich zur Sonne orientiert, so dass bei einem fehlenden Paneel die insgesamt erzeugte Energie um 20 % verringert ist.

Abbildung 10.19 zeigt die Verläufe der durch die drei hier betrachteten Konfigurationen durchschnittlich erzeugten Energie als Funktion der Bahnhöhe und der Inklination. Die Standardkonfiguration zeigt dabei im Gegensatz zu den Konfigurationen mit ausklappbaren Paneelen einen näherungsweise linearen Verlauf, wie in den vorherigen Abschnitten beschrieben. Dabei fällt auf, dass die erzeugte Energie auf hoch inklinierten Orbits relativ ähnlich bei allen drei Konfigurationen ist, während die Konfigurationen mit ausklappbaren Paneelen zu niedrigen Inklinationen hin durchschnittlich





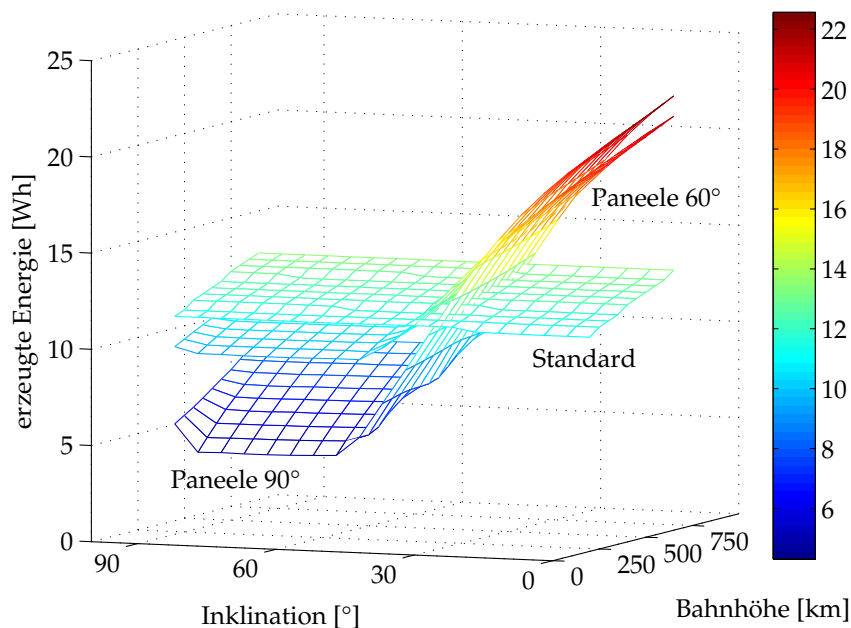
**Abbildung 10.19:** Durchschnittlich durch die Standardkonfiguration und die Konfigurationen mit um  $60^\circ$  bzw.  $90^\circ$  ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.12

deutlich mehr Energie erzeugen, wobei bei der Konfiguration mit um  $90^\circ$  ausgeklappten Paneelen noch etwas höhere Werte erreicht werden. Es gibt also bei Inklinationen zwischen  $0^\circ$  und ungefähr  $85^\circ$  keine Orbits, auf denen die Standardkonfiguration bei durchschnittlicher Sonnenlichtdauer mehr Energie erzeugt, als die Konfigurationen mit ausklappbaren Paneelen. Lediglich auf polnahen Umlaufbahnen kann die Standardkonfiguration je nach RAAN durchschnittlich mehr Energie erzeugen.

Abbildung 10.20 zeigt die minimale durch die drei Konfigurationen insgesamt erzeugte Energie als Funktion der Bahnhöhe und der Inklination. Auch hier zeigt die Standardkonfiguration wieder den linearen Verlauf und die Konfiguration mit um  $90^\circ$  ausgeklappten Paneelen den größten Wertebereich. Es zeigt sich, dass die Standardkonfiguration bezüglich des Worst-Case-Szenarios auf Orbits hoher Inklination mehr Energie erzeugt, als die anderen Konfiguration, wobei insbesondere die Konfiguration mit um  $90^\circ$  ausgeklappten Paneelen hier schlecht abschneidet, da der Betawinkel im schlechtesten Fall nahe  $\pm 90^\circ$  ist, so dass dabei nur sehr wenig Energie erzeugt wird. Auf niedriger inklinierten Orbits sind die Konfigurationen mit ausklappbaren Paneelen jedoch günstiger. Im Vergleich mit Abbildung 10.19 erkennt man darüber hinaus, dass für Inklinationen unter ca.  $20^\circ$  die Konfigurationen mit ausklappbaren Paneelen im schlechtesten Fall sogar mehr Energie erzeugen, als die Standardkonfiguration im Durchschnitt.

Insgesamt lässt sich festhalten, dass die Konfigurationen mit ausklappbaren Paneelen im Durchschnitt auf praktisch allen Orbits mit Inklinationen unter  $90^\circ$  mehr Energie erzeugen, wobei natürlich beachtet werden muss, dass diese Energie während einer





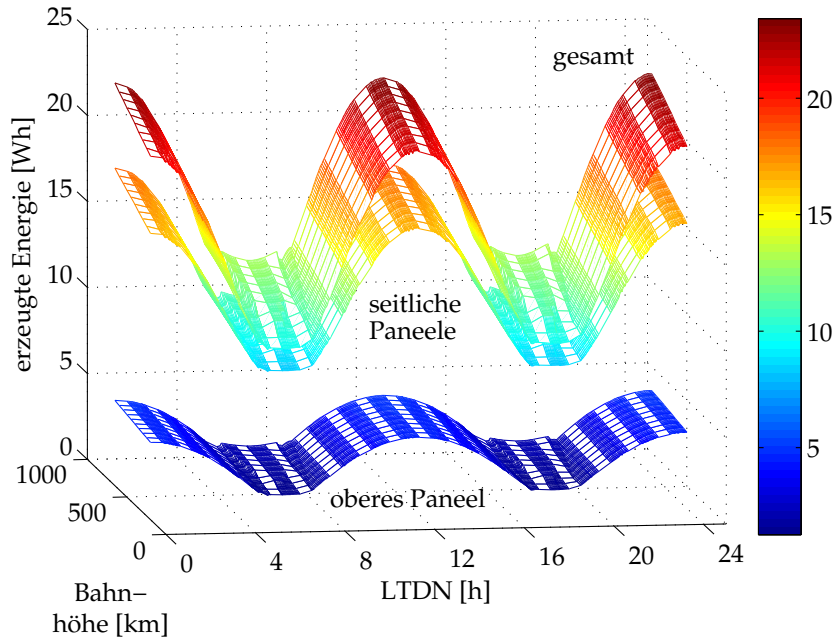
**Abbildung 10.20:** Minimal durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.13

vergleichsweise kurzen Zeit von praktisch allen Paneelen gleichzeitig erzeugt wird, was wie bereits erwähnt hohe Anforderungen an das DEPS stellt. Bei der Standardkonfiguration ist der Verlauf der zur Verfügung stehenden Leistung über die Zeit generell gleichmäßiger (siehe auch Abschnitt 9.1.4). Auf niedrig inklinierten Orbits sind die Konfigurationen mit ausklappbaren Paneelen aufgrund der relativ kurzen Sonnenlichtzeiten deutlich überlegen, während auf höher inklinierten Bahnen die geringere bei diesen Konfigurationen erzeugte Energie im schlechtesten Fall in Betracht gezogen werden muss.

### Sonnensynchrone Orbits

Wie bei der Standardkonfiguration gilt auch hier, dass die erzeugte Energie auf SSOs praktisch nur von der Bahnhöhe und der LTDN abhängt. Bei der während einer durchschnittlich langen Sonnenlichtperiode erzeugten Energie zeigt sich im Gegensatz zur Standardkonfiguration erwartungsgemäß ein ähnlicher Verlauf der erzeugten Energie über die LTDN bei dem oberen Paneel und den seitlichen Paneelen, wie in Abbildung 10.21 dargestellt. Dabei sind die Verläufe der vom oberen Paneel, von den seitlichen Paneelen und der insgesamt erzeugten Energie dargestellt. Es ergibt sich eine größere Spanne der Werte bei verschiedenen LTDN aufgrund der gleichen Lage der Maxima und Minima.

Zur Näherung dieser Verläufe wird wie in Abschnitt 10.4.1 Gleichung (10.10) verwendet. Dabei entspricht hier der Verlauf für das obere Paneel dem für das obere Paneel der Standardkonfiguration, weshalb dieser hier nicht weiter betrachtet wird. Für den



**Abbildung 10.21:** Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um 60° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.14

Verlauf der durch die seitlichen Paneele erzeugten Energie existiert wiederum eine Grenzünergie als Funktion der Bahnhöhe, die nicht unterschritten wird und durch folgende Gleichung angenähert werden kann:

$$E_{\text{grenz,ave,side}}^{\text{min}}(h) = 5,0844 \cdot 10^{-3} \frac{\text{Wh}}{\text{km}} \cdot h + 7,4851 \text{ Wh} \quad (10.17)$$

Die Koeffizienten der Näherungsgleichung für die seitlichen Paneele bei Energien über  $E_{\text{grenz,ave,side}}^{\text{min}}$  sind in Tabelle 10.18 zusammengefasst. Der maximale dabei auftretende Fehler liegt im Bereich der LTDN von 7 h, wo der Übergang zwischen der durch die Näherungsgleichung beschriebenen Energie und der auf den Minimalwert gesetzten Energie stattfindet, und beträgt ca. 3,5 %. Im Durchschnitt ergibt sich ein Fehler von ca. 0,7 %.

Die Verläufe der im schlechtesten Fall erzeugten Energie ähneln denen der durchschnittlich erzeugten Energie aufgrund der ähnlichen Orientierung der Paneele und den geringen Schwankungen der Dauer der Sonnenlichtzeiten auf SSOs bei der hier betrachteten Konfiguration. Die größten Unterschiede ergeben sich auf Dämmerungsortbits durch die dabei auftretenden verhältnismäßig großen Schwankungen der Dauer der Sonnenlichtzeiten und liegen im Bereich von einigen zehntel Wattstunden. Aufgrund dieser geringen Unterschiede werden die Verläufe hier nicht bildlich dargestellt, Abbildung G.15 im Anhang zeigt den Verlauf im zweidimensionalen.

Der Verlauf der durch die seitlichen Paneele erzeugten Energie zeigt hier, ähnlich wie beim oberen Paneel, ein Minimum, das durch folgende Näherungsgleichung beschrie-

$q_1 =$	$1,5064 \cdot 10^1$	[Wh]	$q_{11} =$	$5,0950 \cdot 10^1$	[Wh]
$q_2 =$	$4,8619 \cdot 10^{-3}$	[Wh/km]	$q_{12} =$	$1,6808 \cdot 10^{-2}$	[Wh/km]
$q_3 =$	1,4897	[Wh/h]	$q_{13} =$	$-1,8676 \cdot 10^1$	[Wh/h]
$q_4 =$	$1,2749 \cdot 10^{-6}$	[Wh/km <sup>2</sup> ]	$q_{14} =$	$-7,5410 \cdot 10^{-6}$	[Wh/km <sup>2</sup> ]
$q_5 =$	$-1,1884$	[Wh/h <sup>2</sup> ]	$q_{15} =$	2,4328	[Wh/h <sup>2</sup> ]
$q_6 =$	$-1,5771 \cdot 10^{-9}$	[Wh/km <sup>3</sup> ]	$q_{16} =$	$9,0497 \cdot 10^{-10}$	[Wh/km <sup>3</sup> ]
$q_7 =$	$1,2064 \cdot 10^{-1}$	[Wh/h <sup>3</sup> ]	$q_{17} =$	$-9,3318 \cdot 10^{-2}$	[Wh/h <sup>3</sup> ]
$q_8 =$	$-1,6877 \cdot 10^{-3}$	[Wh/(km·h)]	$q_{18} =$	$-2,3130 \cdot 10^{-3}$	[Wh/(km·h)]
$q_9 =$	$2,8708 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]	$q_{19} =$	$6,9118 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]
$q_{10} =$	$2,6791 \cdot 10^{-4}$	[Wh/(km·h <sup>2</sup> )]	$q_{20} =$	$9,7152 \cdot 10^{-5}$	[Wh/(km·h <sup>2</sup> )]

**Tabelle 10.18:** Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch die seitlichen Paneele der Konfiguration mit um 60° ausgeklappten Paneelen bei durchschnittlicher Gesamtenergie erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien über  $E_{\text{grenz,ave,side}}^{\min}$

ben werden kann:

$$E_{\text{grenz,min,side}}^{\min}(h) = 3,7111 \cdot 10^{-3} \frac{\text{Wh}}{\text{km}} \cdot h + 7,8578 \text{Wh} \quad (10.18)$$

Die zur Näherung in Gleichung (10.10) einzusetzenden Koeffizienten sind in Tabelle 10.19 gegeben. Hier muss allerdings beachtet werden, dass die Funktionswerte der Näherungsgleichung im Bereich konstanter, minimaler Energie nicht wie in den bisherigen Fällen kleiner sind, als  $E_{\text{grenz,min,side}}^{\min}$ . Daher muss die erzeugte Energie auf die Grenzenergie gesetzt werden, wenn die berechnete Energie kleiner ist als  $E_{\text{grenz,min,side}}^{\min}$  oder wenn die LTDN zwischen 5 h und 7 h bzw. zwischen 11 h und 13 h liegt. Die hier auftretenden Fehler sind in der gleichen Größenordnung wie bei der durchschnittlich durch die um 60° ausgeklappten seitlichen Paneele erzeugten Energie (siehe oben).

Für das obere Paneel ist der Verlauf für die minimal erzeugte Gesamtenergie ungleich dem entsprechenden Verlauf bei der Standardkonfiguration, da bei der hier betrachteten Konfiguration die minimale Gesamtenergie während der kürzesten Sonnenlichtzeiten erzeugt wird. Es ergibt sich die folgende Näherungsgleichung für die Grenzenergie als Funktion der Bahnhöhe:

$$E_{\text{grenz,min,top}}^{\min}(h) = 2,1511 \cdot 10^{-3} \frac{\text{Wh}}{\text{km}} \cdot h + 0,4138 \text{Wh} \quad (10.19)$$

Die Faktoren zur Näherung des Verlaufs finden sich in Tabelle 10.20. Bei der Näherung tritt ein maximaler Fehler von ca. 17,4 % im Übergangsbereich der LTDN bei hohen Orbits auf. Der durchschnittliche Fehler beträgt ca. 2,5 %.

Wie auf den Orbits niedrigerer Inklination ergibt sich auch hier die durch die Konfiguration mit um 90° ausgeklappten Paneelen im schlechtesten bzw. durchschnittlichen Fall erzeugte Energie aus den für das obere Paneel der Konfiguration mit um 60° ausgeklappten Paneelen gegebenen Näherungen unter Verwendung der Gleichungen

$q_1 =$	$1,4370 \cdot 10^1$	[Wh]	$q_{11} =$	$1,0220 \cdot 10^2$	[Wh]
$q_2 =$	$1,0682 \cdot 10^{-2}$	[Wh/km]	$q_{12} =$	$2,0857 \cdot 10^{-2}$	[Wh/km]
$q_3 =$	1,1905	[Wh/h]	$q_{13} =$	$- 3,4687 \cdot 10^1$	[Wh/h]
$q_4 =$	$- 1,5398 \cdot 10^{-5}$	[Wh/km <sup>2</sup> ]	$q_{14} =$	$- 7,1287 \cdot 10^{-6}$	[Wh/km <sup>2</sup> ]
$q_5 =$	$- 1,3830$	[Wh/h <sup>2</sup> ]	$q_{15} =$	4,0335	[Wh/h <sup>2</sup> ]
$q_6 =$	$1,2167 \cdot 10^{-8}$	[Wh/km <sup>3</sup> ]	$q_{16} =$	$7,0666 \cdot 10^{-9}$	[Wh/km <sup>3</sup> ]
$q_7 =$	$1,6890 \cdot 10^{-1}$	[Wh/h <sup>3</sup> ]	$q_{17} =$	$- 1,4578 \cdot 10^{-1}$	[Wh/h <sup>3</sup> ]
$q_8 =$	$- 1,0147 \cdot 10^{-3}$	[Wh/(km·h)]	$q_{18} =$	$- 3,2177 \cdot 10^{-3}$	[Wh/(km·h)]
$q_9 =$	$- 2,1473 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]	$q_{19} =$	$- 2,3502 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]
$q_{10} =$	$2,0526 \cdot 10^{-4}$	[Wh/(km·h <sup>2</sup> )]	$q_{20} =$	$1,8383 \cdot 10^{-4}$	[Wh/(km·h <sup>2</sup> )]

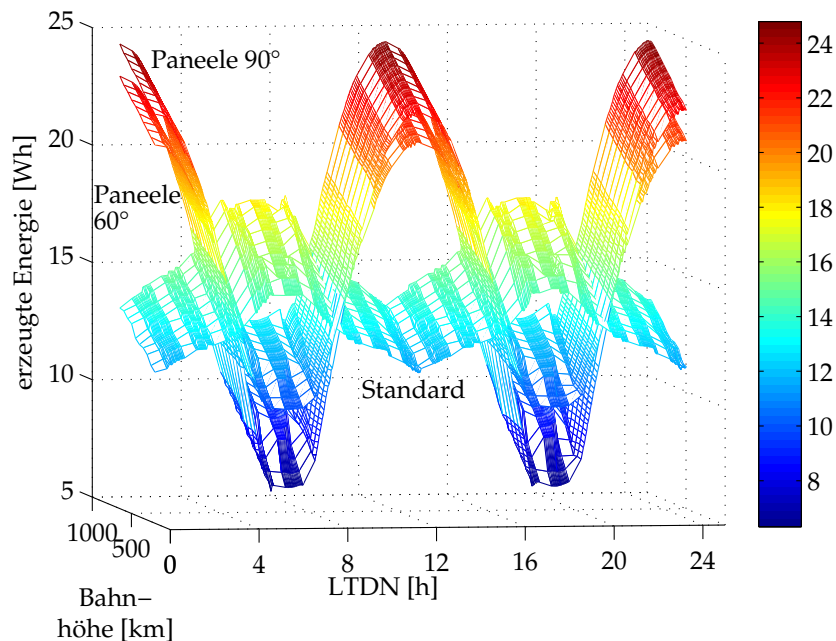
**Tabelle 10.19:** Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch die seitlichen Paneele der Konfiguration mit um 60° ausgeklappten Paneelen bei minimaler Gesamtenergie erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für Energien über  $E_{\text{grenz,min,side}}^{\text{min}}$  und außerhalb der Bereiche der LTDN zwischen 5 h und 7 h sowie zwischen 11 h und 13 h

$q_1 =$	3,4808	[Wh]	$q_{11} =$	8,8412	[Wh]
$q_2 =$	$5,2801 \cdot 10^{-3}$	[Wh/km]	$q_{12} =$	$1,7762 \cdot 10^{-2}$	[Wh/km]
$q_3 =$	$5,1859 \cdot 10^{-1}$	[Wh/h]	$q_{13} =$	$- 5,3207$	[Wh/h]
$q_4 =$	$- 8,3247 \cdot 10^{-6}$	[Wh/km <sup>2</sup> ]	$q_{14} =$	$1,3609 \cdot 10^{-5}$	[Wh/km <sup>2</sup> ]
$q_5 =$	$- 3,7379 \cdot 10^{-1}$	[Wh/h <sup>2</sup> ]	$q_{15} =$	$8,5347 \cdot 10^{-1}$	[Wh/h <sup>2</sup> ]
$q_6 =$	$6,2356 \cdot 10^{-9}$	[Wh/km <sup>3</sup> ]	$q_{16} =$	$- 1,8004 \cdot 10^{-8}$	[Wh/km <sup>3</sup> ]
$q_7 =$	$2,6819 \cdot 10^{-2}$	[Wh/h <sup>3</sup> ]	$q_{17} =$	$- 3,6116 \cdot 10^{-2}$	[Wh/h <sup>3</sup> ]
$q_8 =$	$- 1,3711 \cdot 10^{-3}$	[Wh/(km·h)]	$q_{18} =$	$- 4,6808 \cdot 10^{-3}$	[Wh/(km·h)]
$q_9 =$	$2,7447 \cdot 10^{-7}$	[Wh/(km <sup>2</sup> ·h)]	$q_{19} =$	$1,0021 \cdot 10^{-6}$	[Wh/(km <sup>2</sup> ·h)]
$q_{10} =$	$2,3458 \cdot 10^{-4}$	[Wh/(km·h <sup>2</sup> )]	$q_{20} =$	$1,9846 \cdot 10^{-4}$	[Wh/(km·h <sup>2</sup> )]

**Tabelle 10.20:** Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch das obere Paneel der Konfiguration mit um 60° ausgeklappten Paneelen bei minimaler Gesamtenergie erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien über  $E_{\text{grenz,min,top}}^{\text{min}}$

(10.7) und (10.10) sowie den Koeffizienten aus den Tabellen 10.13 und 10.20. Die Verläufe entsprechen bei den hier angenommenen 14 Solarzellen pro Paneel wiederum denen des oberen Paneels multipliziert mit fünf.

Abbildung 10.22 zeigt die Verläufe der von den drei hier betrachteten Konfigurationen durchschnittlich insgesamt erzeugten Energie als Funktion der Bahnhöhe und der LTDN. Grundsätzlich zeigt sich wie erwartet ein ähnlicher Verlauf bei den beiden Konfigurationen mit ausklappbaren Paneelen, wobei die Werte der Konfiguration mit um 90° ausgeklappten Paneelen einen größeren Bereich abdecken. Außerdem erkennt man, dass die Werte für die Standardkonfiguration in einem deutlich engerem Bereich liegen. Der Einsatz der Standardkonfiguration lohnt sich bei durchschnittlich erzeugter Energie ungefähr bei LTDN zwischen 3,5 h und 8,5 h sowie zwischen 15,5 h und 20,5 h, wobei sich nur eine geringe Abhängigkeit von der Bahnhöhe zeigt. Da die



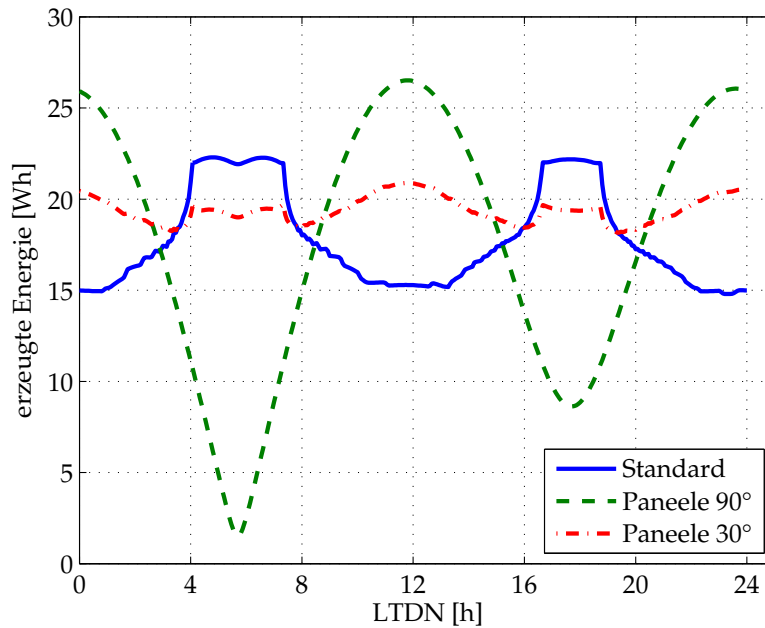
**Abbildung 10.22:** Durchschnittlich durch die Standardkonfiguration und die Konfigurationen mit um  $60^\circ$  bzw.  $90^\circ$  ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der LTDN für SSOs, siehe auch Abbildung G.16

Konfigurationen mit ausklappbaren Paneelen in den Übergangsbereichen ähnliche Werte aufweisen, hängt der Bereich, in dem diese Konfigurationen günstiger bzw. ungünstiger sind, nur unwesentlich von dem Ausklappwinkel ab, sofern dieser zwischen  $60^\circ$  und  $90^\circ$  liegt.

Aufgrund der geringen Schwankungen der Sonnenlichtdauer auf den SSOs weichen die Verläufe der minimal erzeugten Energie unwesentlich von den Verläufen der mittleren erzeugten Energie ab. Dies ändert jedoch nur wenig an den angegebenen Bereichen, in denen die Standardkonfiguration mehr Energie liefert. Daher werden diese Verläufe hier ebenfalls nicht weiter betrachtet. Abbildung G.17 im Anhang zeigt die Verläufe in zweidimensionaler Ansicht.

In [2] wird ein weiterer Vergleich zu einer Konfiguration, bei der die Paneele um  $30^\circ$  aus der ursprünglichen Lage ausgelenkt fest angebracht sind, angestellt, was einen maximalen, sinnvoll zu realisierenden Ausklappwinkel bei einer starren Aufhängung der Paneele darstellt. Eine solche Konfiguration war ursprünglich für die AISat-Mission angedacht. Als Teil dieser Arbeit wurde für SSOs verschiedener LTDN in 650 km Höhe ein beliebiger Orbit im Oktober betrachtet und für diesen Fall die erzeugte Energie der Standardkonfiguration, der Konfiguration mit um  $90^\circ$  ausgeklappten Paneelen sowie der Konfiguration mit um  $30^\circ$  verkippten Paneelen bestimmt. Die Ergebnisse der Rechnungen sind hier als weiterer Vergleich beispielhaft wiedergegeben.

Abbildung 10.23 zeigt die Verläufe über die LTDN. Man erkennt dabei zunächst den



**Abbildung 10.23:** Vergleich der durch die Standardkonfiguration, die Konfiguration mit um 90° ausgeklappten Paneelen und die Konfiguration mit um 30° verkippten Paneelen erzeugten Energie über eine Sonnenlichtperiode im Oktober auf SSOs verschiedener LTDN in 650 km Höhe

oben genannten Bereich der LTDN, in dem die Standardkonfiguration mehr Energie erzeugt, als die Konfiguration mit um 90° ausgeklappten Paneelen. Bei der dritten Konfiguration zeigt sich, dass dieser Bereich etwas kleiner wird, die Standardkonfiguration also nur noch bei LTDN zwischen 4 h und 8 h bzw. zwischen 16 h und 20 h überlegen ist.

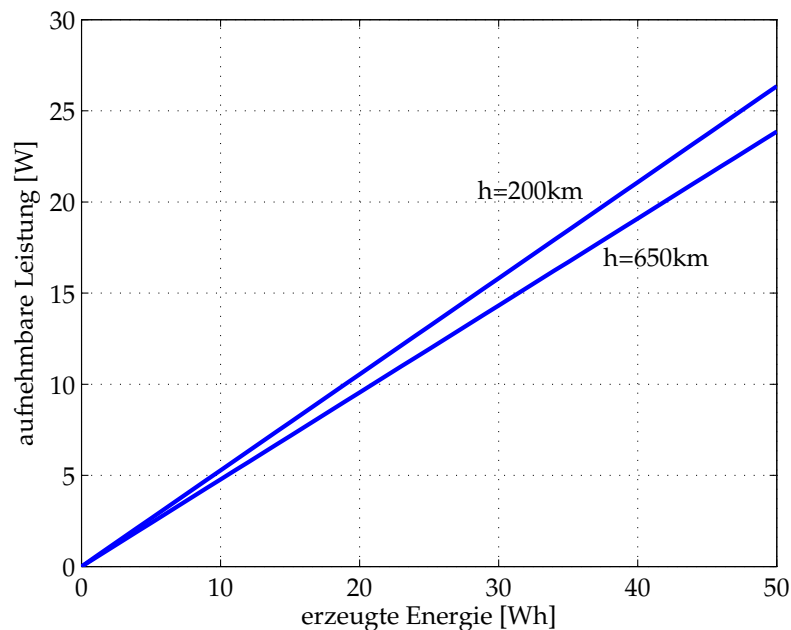
## 10.5 Aufnehmbare Leistung als Funktion der erzeugten Energie

Wie bereits erwähnt kann aus der bisher betrachteten über eine Sonnenlichtphase erzeugten Energie mit Hilfe von Gleichung (4.12) die Leistung bestimmt werden, die von dem Satellitensystem maximal aufgenommen werden kann, damit die Batterie während der Sonnenlichtphase wieder voll geladen wird. Dabei wird allerdings eine konstante Leistungsaufnahme über einen kompletten Orbit und ein dem bei der AISat-Mission verwendeten DEPS ähnliches System vorausgesetzt. Sind also die Sonnenlicht- bzw. die Eklipsendauer sowie die von den Paneelen erzeugte Energie bekannt, kann damit diese Leistung abgeschätzt werden. Es bietet sich folgende Vereinfachung an, mit der  $P_{auf}$  konservativ abgeschätzt werden kann:

$$P_{auf} = \frac{E_{pan} \cdot \eta_{BCR} \cdot \eta_{bus}}{RF \cdot t_U} \quad (10.20)$$

Mit dieser vereinfachten Gleichung ergibt sich ein maximaler Fehler von ca. 6 % gegenüber Gleichung (4.12). Damit ist aber die aufnehmbare Leistung nur noch eine Funk-

tion der erzeugten Energie und der Bahnhöhe unter Berücksichtigung von Gleichung (3.2). Abbildung 10.24 zeigt die Verläufe der aufnehmbaren Leistung als Funktion der erzeugten Energie nach Gleichung (10.20) für die Bahnhöhen 200 km und 650 km. Die Verläufe der anderen Bahnhöhen werden hier der Übersichtlichkeit halber weggelassen.



**Abbildung 10.24:** Von dem Satellitensystem bei vollständiger Aufladung der Batterie aufnehmbare Leistung als Funktion der erzeugten Energie über eine Sonnenlichtperiode für verschiedene Bahnhöhen

Mit Hilfe dieses Diagramms lässt sich aus den in den vorigen Abschnitten gegebenen Werten für die erzeugte Energie die entsprechend mögliche Leistungsaufnahme der Satelliteninstrumente abschätzen. Die Abhängigkeit von der Bahnhöhe ist dabei nicht als Änderung eines Wirkungsgrades zu interpretieren. Mit steigender Bahnhöhe steigt die Umlaufdauer und damit auch die während eines Orbits benötigte Energie zur Versorgung des Satellitensystems mit elektrischer Energie. Auf höheren Orbits muss daher während einer Sonnenlichtperiode mehr Energie als auf niedrigeren Orbits erzeugt werden, damit das Satellitensystem dauerhaft die gleiche Leistung aufnehmen kann.

Eine auf verschiedenen Orbits erzeugte Durchschnittsenergie der Standardkonfiguration von ca. 14 Wh würde nach dieser konservativen Abschätzung also dauerhaft für eine Leistungsaufnahme von ca. 7 W reichen. Die Konfigurationen mit ausklappbaren Paneelen können mit Standardpaneelen ausgestattet auf günstigen Orbits bis zu 25 Wh erzeugen, womit eine durchschnittliche Leistungsaufnahme von ca. 12 W möglich wird. Setzt man Paneele mit mehr Solarzellen zu einem günstigen Zeitpunkt im Jahr ein, steigen dieser Werte natürlich entsprechend. Mit den angegebenen Werten ist es möglich, den bestmöglichen Fall zu betrachten. Geht man von einer maximalen von

der Sonne eingestrahlen Leistung von  $1461,2 \frac{W}{m^2}$  und von 28 Solarzellen auf dem oberen Paneel und jeweils 24 Zellen auf den seitlichen Paneelen aus, ergibt sich auf einem polaren Orbit in 650 km Höhe bei der Konfiguration mit um  $90^\circ$  ausgeklappten Paneelen im besten Fall eine erzeugte Energie von knapp 47 Wh. Damit könnte theoretisch eine Leistungsaufnahme von über 20 W ermöglicht werden. Allerdings muss dabei wie bereits erwähnt das DEPS an den zeitlichen Verlauf der Erzeugung der Energie angepasst werden.

## 10.6 Bodenstationskontakt

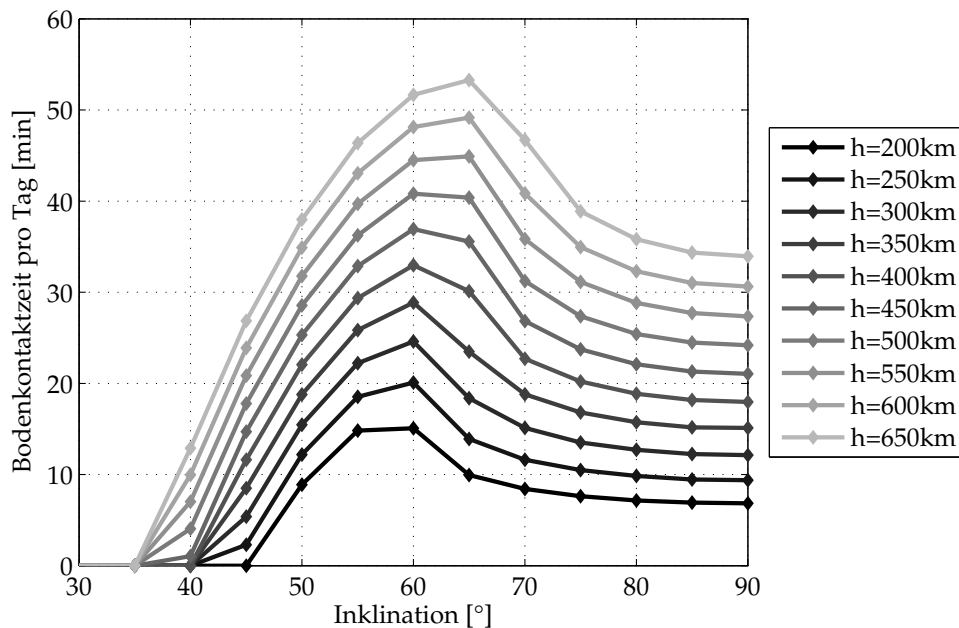
Für die zukünftigen CLAVIS-Missionen werden hier die Bodenkontaktzeiten auf verschiedenen Orbits dargestellt. Dabei werden die Bodenstation am DLR Institut für Raumfahrtssysteme in Bremen (ca.  $53^\circ$  Nord) und die DLR Bodenstation in Inuvik in Kanada (ca.  $68^\circ$  Nord) berücksichtigt (siehe Abschnitt 5.1). Grundsätzlich zeigt sich auf den hier betrachteten LEOs über einen Zeitraum von mindestens einigen Tagen eine praktisch konstante Bodenkontaktzeit pro Zeiteinheit, wie auch in Abbildung 9.14 für die AISat-Mission dargestellt ist. Dabei muss beachtet werden, dass die Inklination des Orbits des Satelliten groß genug sein muss, damit der Satellit für eine Bodenstation, die nicht am Äquator liegt, überhaupt sichtbar wird. Dieser konstante Wert ergibt sich aufgrund der geringen Umlaufzeit des Satelliten im Vergleich zur Erddrehung. Die Bodenspur des Satelliten füllt im allgemeinen Fall mit der Zeit annähernd die komplette Erdoberfläche innerhalb eines Bandes, das in seiner Breite durch die Inklination des Satelliten begrenzt ist. Damit kann jede Bodenstation, die innerhalb oder gerade außerhalb dieses Bandes liegt, regelmäßig Kontakt zu dem Satelliten aufnehmen.

Da die Bodenstation in Bremen näher am Äquator liegt, wird die Inklination des Orbits zukünftiger CLAVIS-Missionen durch die geografische Lage dieser Bodenstation begrenzt. Auf Inklinationen unterhalb von ca.  $40^\circ$  wird der Satellit von der Bodenstation aus gesehen zu keinem Zeitpunkt über dem Horizont erscheinen. Soll CLAVIS in einen solchen Orbit gebracht werden, ist eine weitere Bodenstation in niedrigeren Breiten nötig.

Da die Bodenkontaktdauer pro Zeiteinheit wie erwähnt mittelfristig betrachtet konstant ist, wird hier die Bodenkontaktzeit pro Tag als Kriterium zur Beurteilung der verschiedenen Orbits herangezogen. Abbildung 10.25 zeigt die Bodenkontaktdauer pro Tag in Abhängigkeit der Inklination für prograde Orbits für verschiedene Bahnhöhen unter Berücksichtigung der minimalen Elevation von  $10^\circ$ . Dabei haben wiederum die Startrektaszension und der Startzeitpunkt auf längere Sicht praktisch keinen Einfluss.

Man erkennt zunächst, dass die Bodenkontaktzeit grundsätzlich mit der Bahnhöhe steigt, was zum einen auf die geringere Kreisbahngeschwindigkeit in größeren Höhen zurückzuführen ist und zum anderen auf die Tatsache, dass sich ein Satellit in größerer Höhe zu einem gegebenen Zeitpunkt von einem größeren Anteil der Erdoberfläche aus gesehen über dem Horizont befindet. Damit steigt auch die Zeitspanne, während der ein Satellit bei einem Überflug von einem festen Punkt auf der Erdoberfläche aus





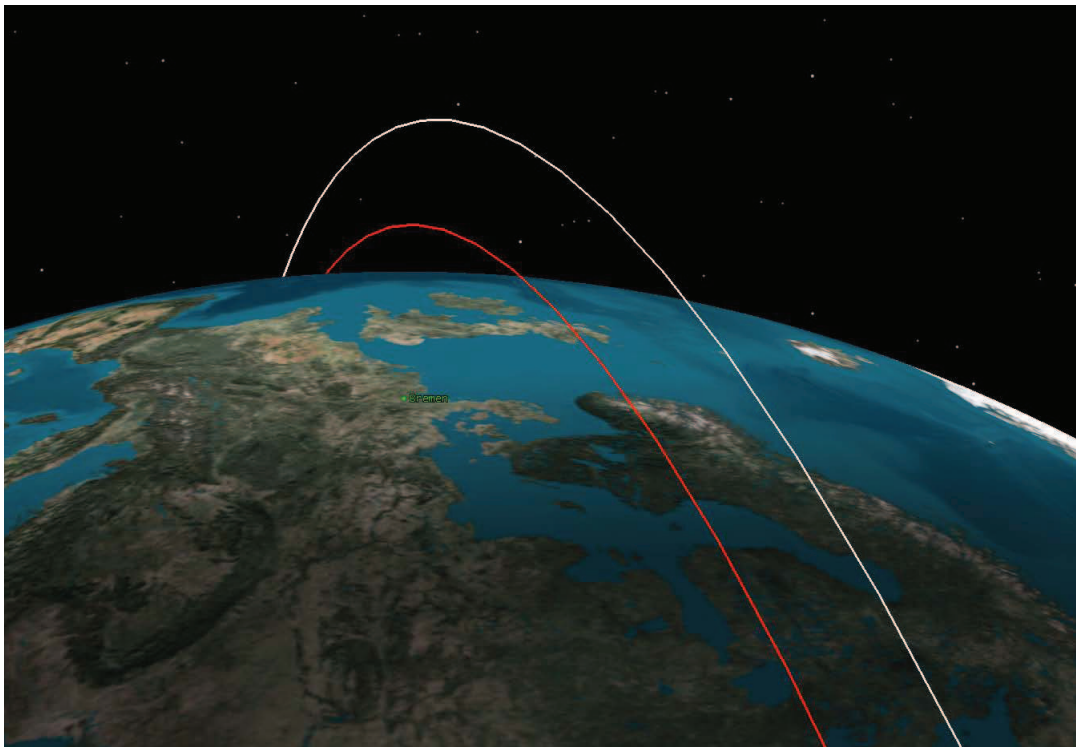
**Abbildung 10.25:** Durchschnittlich mögliche Bodenkontaktdauer mit der Bodenstation in Bremen pro Tag als Funktion der Inklination für verschiedene Bahnhöhen

gesehen werden kann.

Bei geringen Inklinationen ist die Bodenkontaktzeit wie bereits erwähnt gleich null. Zu höheren Inklinationen hin steigt die Bodenkontaktzeit zunächst an und erreicht ihr Maximum, je nach Bahnhöhe, bei Inklinationen, die etwas über der nördlichen Breite der Bodenstation liegen. Entspräche die Inklination der Breite der Bodenstation, wäre der Satellit nur unter optimalen Umständen von der Bodenstation aus gesehen im Zenit, was aufgrund der Relativbewegungen nur sehr selten vorkommt. Ist die Inklination jedoch etwas größer, gibt es stets mehrere aufeinander folgende Umläufe, während denen der Satellit eine hohe Elevation erreicht und damit für eine lange Zeit von der Bodenstation aus sichtbar ist, weshalb sich bei diesen Inklinationen das Maximum der Bodenkontaktzeiten befindet.

Das Maximum der Bodenkontaktzeiten liegt bei höheren Orbits bei höheren Inklinationen, da auf niedrigen Orbits geringere Abweichungen zwischen der nördlichen Breite der Bodenstation und der Inklination zu einer starken Reduktion der maximalen Elevation des Satelliten während eines Überfluges führen. Die entsprechende Geometrie ist in Abbildung 10.26 verdeutlicht, in der zwei Orbits in 200 km und 650 km Höhe mit einer Inklination von jeweils  $60^\circ$  sowie die Bodenstation in Bremen dargestellt sind. Die dargestellten Orbits sind abgesehen von der Bahnhöhe identisch. Man erkennt, dass die maximale Elevation des Satelliten auf dem höheren Orbit bei einem Überflug deutlich größer ist, als bei einem Satelliten auf dem niedrigeren Orbit.

Auf Orbits höherer Inklination sind Überflüge, während der der Satellit eine Elev-

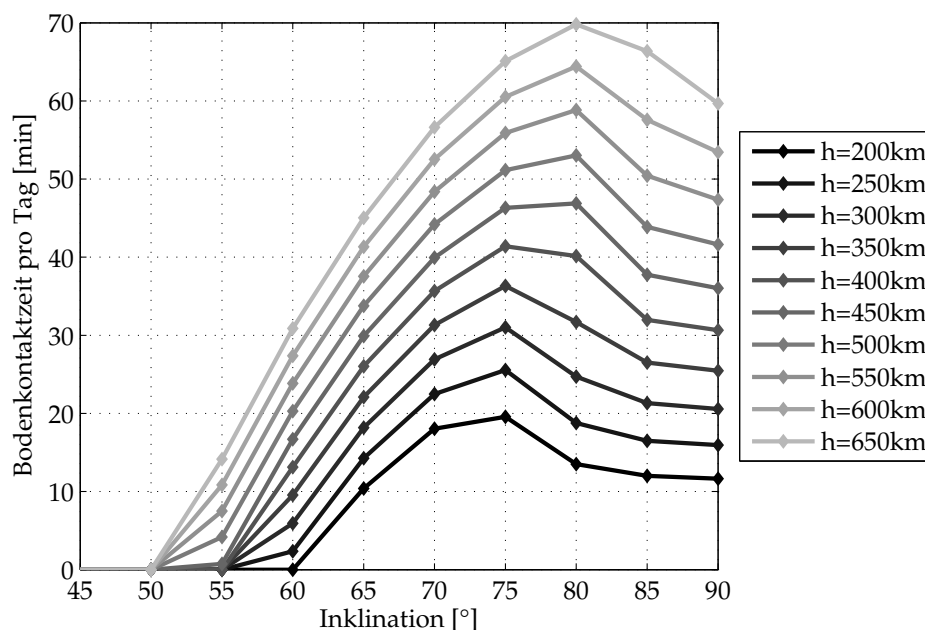


**Abbildung 10.26:** Vergleich der Geometrie des möglichen Bodenkontakts zwischen der Bodenstation in Bremen und Satelliten auf Orbits in 200 km und 650 km Höhe bei einer Inklination von  $60^\circ$

tion nahe  $90^\circ$  erreicht, wiederum relativ selten, da die Bodenspur des Satelliten eher in Nord-Süd-Richtung verläuft. Damit verringert sich auch die durchschnittliche Bodenkontaktzeit während eines Tages hin zu höheren Inklinationen. Bei diesen Betrachtungen ist es unerheblich, ob prograde oder retrograde Orbits betrachtet werden, da allein die Kombination aus der Umlaufdauer und der Erddrehung den Verlauf der durchschnittlichen Bodenkontaktzeiten bewirkt. Die unterschiedliche Richtung der  $J_2$ -Störung hat dabei keinen merklichen Einfluss. Der Verlauf für Inklinationen zwischen  $90^\circ$  und  $180^\circ$  entspricht dem für niedrigere Inklinationen dargestellten, gespiegelt an einer Vertikalen bei  $90^\circ$  Inklination. Für SSOs können die Werte der Bodenkontaktzeit daher aus dem gegebenen Diagramm bei der entsprechenden Bahnhöhe und der entsprechenden Inklination kleiner  $90^\circ$ , gegeben durch  $i = 180^\circ - i_{SSO}$  abgelesen werden.

Abbildung 10.27 zeigt analog zu Abbildung 10.25 den Verlauf der durchschnittlichen Bodenkontaktzeit pro Tag zur Bodenstation in Inuvik als Funktion der Inklination für verschiedene Bahnhöhen. Es zeigen sich sehr ähnliche Verläufe wie bei der Bodenstation in Bremen, wobei hier die Inklination mindestens  $55^\circ$  bis  $60^\circ$  betragen muss, damit überhaupt ein Kontakt zu dieser Bodenstation möglich ist. Aufgrund der nördlicheren Lage dieser Bodenstation verschiebt sich das Maximum der durchschnittlichen Bodenkontaktzeit hin zu höheren Inklinationen. Für SSOs gilt die gleiche Abschätzung wie bei der Bodenstation in Bremen.

Grundsätzlich ist die durchschnittliche Bodenkontaktzeit zur Bodenstation in Inuvik bei ausreichend hoch inklinierten Orbits größer als zur Bodenstation in Bremen. Auch

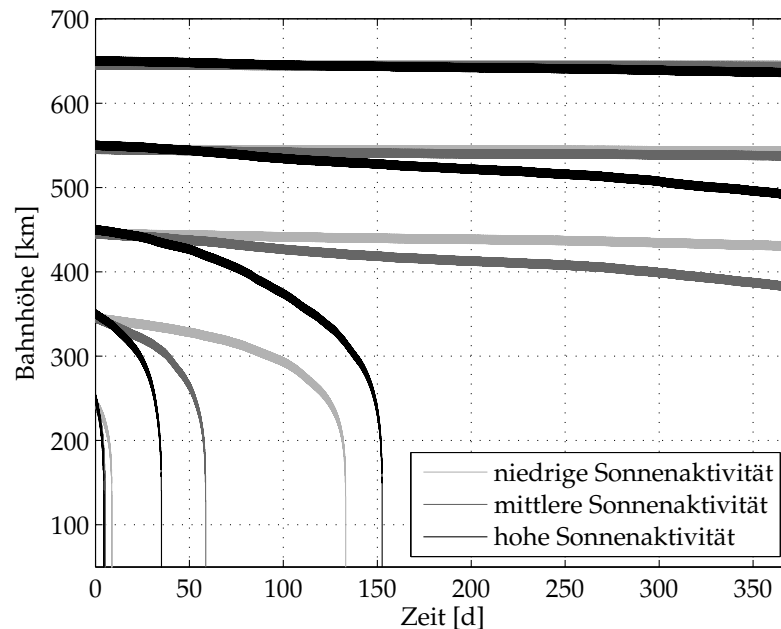


**Abbildung 10.27:** Durchschnittlich mögliche Bodenkontaktdauer mit der Bodenstation in Inuvik pro Tag als Funktion der Inklination für verschiedene Bahnhöhen

dies erklärt sich durch die Breiten, in denen sich die Stationen befinden. Gäbe es eine Bodenstation an einem Pol, könnte diese bei jedem Umlauf Kontakt zu einem Satelliten auf einer polnahen Umlaufbahn aufnehmen, während eine äquatornahe Bodenstation nur auf wenigen Orbits in Folge einen Funkkontakt aufbauen könnte. Grundsätzlich gilt, dass die Anzahl der aufeinander folgenden Umläufe eines Satelliten ausreichender Inklination, während der ein Kontakt zu einer Bodenstation hergestellt werden kann, mit steigender Entfernung der Bodenstation vom Äquator zunimmt. Damit steigt auch die durchschnittliche Bodenkontaktzeit pro Tag zu einer Bodenstation.

## 10.7 Evolution der Bahnhöhe über ein Jahr

Für die bisher vorgestellten Simulationsergebnisse der zukünftigen CLAVIS-Missionen wurden stets der Einfluss der Atmosphäre sowie andere Störeinflüsse mit Ausnahme der  $J_2$ -Störung vernachlässigt. Die gegebenen Richtwerte für die verschiedenen Bahnhöhen gelten dementsprechend nicht streng für eine ganze Mission über ein Jahr, sondern nur für den zeitlichen Abschnitt einer Mission, während dem sich der Satellit auf der entsprechenden Bahnhöhe befindet. Um diese zeitliche Entwicklung für einen Start in eine bestimmte Bahnhöhe abschätzen zu können, werden hier abschließend mit dem HPOP simulierte Verläufe der Bahnhöhe über ein Jahr betrachtet. Dabei muss beachtet werden, dass sich auf SSOs bei Verringerung der Bahnhöhe durch die Atmosphäre die Geschwindigkeit der Knotenliniendrehung aufgrund der  $J_2$ -Störung ändert, da diese von der Bahnhöhe abhängt. Kurzfristig betrachtet ändert sich dadurch die LTDN, längerfristig kann ein solcher Orbit dann nicht mehr als sonnensynchron bezeichnet werden.



**Abbildung 10.28:** Verlauf der Bahnhöhe von CLAVIS über ein Jahr unter Berücksichtigung der Störeinflüsse für Missionen auf verschiedenen Bahnhöhen jeweils bei hoher, niedriger und mittlerer Sonnenaktivität

Abbildung 10.28 zeigt den Verlauf der Bahnhöhe über ein Jahr für verschiedene Startbahnhöhen und zu Zeiten verschiedener Sonnenaktivität. Die dabei verwendeten Einstellungen des HPOP sind in Abschnitt 8.2.3 beschrieben. Für die hohe Sonnenaktivität werden Missionen über das Jahr 1991 simuliert, für die mittlere Aktivität Missionen über das Jahr 1993 und für die geringe Aktivität Missionen über das Jahr 1996 [16]. Ein zur Simulation verwendetes MATLAB-Skript findet sich im Anhang F.3.

Man erkennt, dass der Satellit von niedrigen Bahnhöhen innerhalb einiger Tage wieder eintritt, während auf der hier als Maximum betrachteten Bahnhöhe von 650 km über ein Jahr nur eine sehr geringe Abnahme der Höhe festgestellt werden kann. Darüber hinaus wird klar, dass die Sonnenaktivität einen großen Einfluss auf die Entwicklung der Bahnhöhe hat. Bei hoher Sonnenaktivität tritt der Satellit nach der hier dargestellten Rechnung zum Beispiel von einer anfänglichen Bahnhöhe von 450 km innerhalb eines halben Jahres wieder ein, während sich die Bahnhöhe bei niedriger Sonnenaktivität aufgrund der geringeren Ausdehnung der Atmosphäre über ein Jahr nur um einige Kilometer verringert. Daraus wird auch ersichtlich, dass die hier gegebenen Simulationsergebnisse nur als sehr grobe Anhaltspunkte verwendet werden können.

## 10.8 Fazit

In diesem Kapitel wurden die Ergebnisse der für zukünftige CLAVIS-Missionen durchgeführten Simulationen ausführlich beschrieben und entsprechende Nä-

herungsgleichungen angegeben. Dies beinhaltet zunächst Anhaltswerte für die minimalen, durchschnittlichen und maximalen Sonnenlichtzeiten auf den verschiedenen betrachteten Orbits (Abschnitt 10.2). Da die von CLAVIS erzeugte Energie wie gezeigt praktisch nur eine Funktion der Bahnhöhe und der Sonnenlichtdauer sowie der betrachteten Konfiguration ist, lässt sich die während einer bestimmten Sonnenlichtperiode erzeugte Energie aus den Anhaltswerten für die Sonnenlichtzeiten für einen gegebenen Orbit oder mit Hilfe von STK gewonnenen Angaben zur Sonnenlichtdauer für einen konkreten Orbit abschätzen (Abschnitt 10.3). In einem weiteren Schritt lassen sich für die hier betrachteten Orbits Richtwerte für die erzeugte Energie als Funktion der Bahnparameter (Bahnhöhe, Inklination, LTDN) in einem durchschnittlichen und dem schlechtesten Fall angeben (Abschnitt 10.4). Mit den dazu jeweils gegebenen Funktionen, mit denen sich auch Konfigurationen mit weniger Paneelen oder Paneele mit unterschiedlich vielen Solarzellen berücksichtigen lassen, kann eine grobe Auslegung zukünftiger CLAVIS-Missionen bezüglich des Energieversorgungssystems durchgeführt werden. Im Anhang H findet sich eine Übersicht, in der für die verschiedenen Berechnungen die jeweils benötigten Gleichungen und Tabellen sowie die relevanten Abbildungen zusammengefasst sind.

Dabei zeigte sich, dass unabhängig von der Inklination auf höheren Orbits grundsätzlich mehr Energie erzeugt wird, was sich unter anderem durch die längere Umlaufdauer erklärt. Damit wird allerdings auf jedem Umlauf auch mehr Energie benötigt, so dass bei gleichbleibender erzeugter Energie auf niedrigeren Orbits eine höhere Leistungsaufnahme des Satelliten dauerhaft sichergestellt werden kann (Abschnitt 10.5). Insgesamt überwiegt jedoch der erste Effekt, so dass auf höheren Orbits grundsätzlich eine höhere elektrische Leistung zum Betrieb der Instrumente des Satelliten bereitgestellt werden kann. Darüber hinaus nimmt auch die mögliche Bodenkontaktzeit mit der Bahnhöhe zu (Abschnitt 10.6) und der Einfluss der Atmosphäre ist geringer, so dass der Satellit über eine längere Zeit in der gleichen Bahnhöhe bleibt und sich somit die Bedingungen für Messungen oder Beobachtungen nicht so stark ändern. Allerdings sinkt mit zunehmender Höhe die Qualität der Funkverbindung zwischen Satellit und Bodenstation.

Auf prograden Orbits hängt die optimale Inklination von der Konfiguration ab. Die Konfigurationen mit ausklappbaren Paneelen erzeugen besonders auf niedrig inklinierten Bahnen viel Energie, wo die Sonnenlichtzeiten relativ kurz sind. Die Standardkonfiguration ist besonders auf hochinklinierten Orbits zu bevorzugen, da hier längere Sonnenlichtperioden auftreten. Mit Ausnahme von Orbits mit Inklinationen nahe  $90^\circ$  können jedoch die Konfigurationen mit ausklappbaren Paneelen durchschnittlich mehr Energie erzeugen als die Standardkonfiguration. Allerdings schwanken die Werte bei den Konfigurationen mit ausklappbaren Paneelen stärker, im Worst-Case erzeugen diese bei Inklinationen zwischen ca.  $35^\circ$  und  $90^\circ$  weniger Energie als die Standardkonfiguration. Grundsätzlich ist auch stets zu beachten, dass der Verlauf der durch die Standardkonfiguration zur Verfügung gestellten Leistung über eine Sonnenlichtperiode gleichmäßiger ist. Bei den Konfigurationen mit ausklappbaren Paneelen wird viel Energie in einem kurzen Zeitraum erzeugt, so dass das DEPS entsprechend angepasst werden muss. Auf niedrigen Inklinationen gibt es darüber hinaus keine Möglichkeit eines Funkkontakts zu den zur Zeit in

Frage kommenden Bodenstationen, so dass für eine solche Mission eine zusätzliche äquaturnahe Bodenstation verwendet werden müsste. Bei höheren Inklinationen muss beachtet werden, dass der Satellit über Zeiträume von mehreren Tagen bis Wochen im direkten Sonnenlicht sein kann, was zu einer starken Aufheizung und damit zum Ausfall einiger Instrumente führen könnte.

Auf SSOs beeinflusst vor allem die LTDN die Auswahl der Konfiguration. Auch hier gilt, dass die Standardkonfiguration bei längeren vorherrschenden Sonnenlichtperioden eingesetzt werden sollte, also auf Dämmerungsortbits, während die Konfigurationen mit ausklappbaren Paneelen eher bei LTDN von 12 h bzw. 24 h von Vorteil für die erzeugte Energie sind. Die Standardkonfiguration ist bei gleicher Anzahl der Solarzellen pro Paneel bei LTDN zwischen 3,5 h und 8,5 h sowie zwischen 15,5 h und 20,5 h zu bevorzugen. Aufgrund der relativ konstanten Dauer der Sonnenlichtperioden treten hier auch keine großen Schwankungen zwischen schlechtestem Fall und durchschnittlichen Fall auf. Auch in Hinsicht auf Bodenkontaktzeiten ergeben sich bei den hier betrachteten SSOs keine nennenswerten Unterschiede.

# Kapitel 11

## Zusammenfassung

In der vorliegenden Arbeit wurden für den am Institut für Raumfahrtssysteme des DLR entwickelten Nanosatelliten CLAVIS die von den Solarpaneelen erzeugte Energie sowie die möglichen Bodenkontaktzeiten mit Hilfe der Software STK analysiert. Dabei wurde zum einen die erste konkrete Mission betrachtet, die unter dem Namen AISat läuft. Der Satellit ist dabei quaderförmig, mit Solarpaneelen an fünf von sechs Seiten und trägt eine Nutzlast mit verschiedenen Antennen an der sechsten, die in Richtung Nadir zeigen soll. Eine längere Helixantenne dient dabei zur Gravitationsgradientenstabilisierung. Zum anderen wurden mögliche zukünftige Missionen untersucht, wobei für CLAVIS Umlaufbahnen bis zu einer Bahnhöhe von 650 km vorgesehen sind. Hierbei wurde eine Standardkonfiguration betrachtet, die ebenfalls quaderförmig ist und eine ähnliche Anordnung der Solarpaneele besitzt, wie die AISat-Konfiguration und ebenfalls mit der Seite ohne Solarpaneel in Richtung Nadir ausgerichtet werden soll. Darüber hinaus wird zur Zeit ein Ausklappmechanismus entwickelt, mit dem die seitlichen Paneele in Richtung des oberen Paneels ausgeklappt werden können. Hier wurden Konfigurationen mit den Ausklappwinkeln  $60^\circ$  und  $90^\circ$ , also einer Ausrichtung aller Solarpaneele in Richtung Zenit, betrachtet.

Zu Beginn der Arbeit wurden die Einflüsse auf die Eklipsenzeiten der hier stets betrachteten kreisförmigen LEOs analysiert. Dabei hat der Betawinkel, also der Winkel zwischen Sonneneinstrahlung und Orbitalebene, der auf SSOs näherungsweise konstant ist, den größten Einfluss. Anschließend wurden das Energieversorgungssystem, das Kommunikationssystem sowie das Lageregelungssystem und das Lageverhalten von CLAVIS in Hinblick auf die folgenden Simulationen beschrieben. Auf Grundlage dieser theoretischen Überlegungen wurden Simulationsrechnungen unter vereinfachenden Annahmen durchgeführt, wobei MATLAB-Skripte verwendet wurden, mit denen Szenarios in STK definiert und die gewünschten Daten aus STK gewonnen und ausgewertet wurden.

Die AISat-Mission soll auf einen SSO in 650 km Höhe mit einer LTDN von 10 h gebracht werden, auf dem eine Sonnenlichtperiode ungefähr 65 min dauert. Im Laufe der Mission von einem Jahr schwankt die erzeugte Energie pro Sonnenlichtperiode nur um wenige Prozent und beträgt unter Berücksichtigung der hier getroffenen konservativen Annahmen ca. 12 Wh. Durch Taumelbewegungen um die Nadirrichtung mit Amplituden von wenigen Grad ergibt sich eine maximale Reduktion bzw.

Erhöhung der Energie um ca. 2 %, je nach Lage beim Eintritt in das und Austritt aus dem Sonnenlicht. Die erzeugte Energie reicht aus, um die vom Satelliten maximal aufgenommene Leistung von 4,29 W dauerhaft bereitzustellen. Der vorhandene Spielraum zwischen erzeugter und benötigter Energie beträgt unter Berücksichtigung von Verlusten auch im schlechtesten Fall noch 45 %. Auch die DoD liegt mit maximal 7,5 % deutlich unter dem Maximalwert von 20 %. Ähnliches gilt für die Laderate, die die Batteriekapazität nicht überschreiten sollte und maximal ein Fünftel dieses Wertes erreicht. Die Energieversorgung für die AISat-Mission ist also inklusive einer großzügigen Marge sichergestellt. Dauerhaft wäre sogar eine Leistungsaufnahme des Satellitensystems von ca. 6 W möglich. Zusätzlich wurde der Fall betrachtet, dass sich der Satellit mit der Nutzlast in Richtung Zenit ausrichtet, was zu einer Reduktion der erzeugten Energie um ungefähr die Hälfte führen würde. Damit stünde auch in diesem Fall genug Energie für einen Betrieb der wichtigsten Satellitensysteme zur Verfügung.

Bodenkontakt zu der Bodenstation am DLR in Bremen ist während der AISat-Mission unter den getroffenen Annahmen ungefähr 35 Minuten pro Tag möglich. Ein ausreichendes Träger-zu-Rausch-Verhältnis von über 10 dB ist während einer um ca. 15 % kürzeren Zeit gegeben, also während ungefähr einer halben Stunde pro Tag. Die maximale Länge einer einzelnen Bodenkontaktperiode tritt auf, wenn der Satellit auf seiner Bahn von der Bodenstation aus gesehen den Zenit erreicht und ist ca. neun Minuten lang.

Für die zukünftigen CLAVIS-Missionen wurde zunächst die minimale, durchschnittliche und maximale Dauer der Sonnenlichtperioden auf den verschiedenen möglichen Orbits über Missionen von einem Jahr betrachtet. Die maximale Dauer wird dabei als die Dauer der längsten Sonnenlichtzeit, die kürzer ist als ein Umlauf, definiert. Anschließend wurden die von den verschiedenen Konfigurationen erzeugte Energie als Funktion der Dauer der Sonnenlichtperiode sowie der Bahnhöhe und schließlich Anhaltswerte (der schlechteste Fall und ein durchschnittlicher Fall) der erzeugten Energie als Funktion der Bahnparameter betrachtet und entsprechende Näherungsgleichungen angegeben. Damit kann für einen gegebenen Orbit einer zukünftigen CLAVIS-Mission die auf den verschiedenen Konfigurationen zur Verfügung stehende Energie abgeschätzt werden.

Auf Orbits mit Inklinationen unter  $90^\circ$  hängen die Sonnenlichtzeiten hauptsächlich von der Bahnhöhe und der Inklination ab. Die längsten Sonnenlichtperioden treten dabei im Allgemeinen auf hochinklinierten Orbits großer Bahnhöhe auf. Auf SSOs ist die Dauer der Sonnenlichtzeiten eine Funktion der Bahnhöhe und der LTDN. Es ergeben sich lange Sonnenlichtzeiten auf Dämmerungsortbits und kurze Sonnenlichtzeiten bei LTDN von 12 h oder 24 h.

Für die verschiedenen Konfigurationen hängt die erzeugte Energie direkt von der Bahnhöhe und der Sonnenlichtdauer ab, wobei auf höheren Orbits grundsätzlich mehr Energie erzeugt wird. Bei längeren Sonnenlichtperioden ist der Betawinkel groß, so dass insbesondere die seitlichen Paneele der Standardkonfiguration viel Energie erzeugen. Bei kürzeren Sonnenlichtperioden ist der Betawinkel klein, so dass das



obere Paneel viel Energie erzeugt. Daraus ergibt sich, dass die Standardkonfiguration während längerer Sonnenlichtperioden mehr Energie und die Konfigurationen mit ausklappbaren Paneelen während kürzerer Sonnenlichtperioden mehr Energie erzeugen. Insgesamt können die Konfigurationen mit ausklappbaren Paneelen auf geeigneten Orbits deutlich mehr Energie erzeugen als die Standardkonfiguration. Dabei muss allerdings beachtet werden, dass dabei die Energie über einen sehr kurzen Zeitraum erzeugt wird, da alle Paneele eine ähnliche Ausrichtung zur Sonne besitzen. Dies stellt hohe Ansprüche an das Energieversorgungssystem, das für diesen Fall entsprechend angepasst werden muss.

Abschließend wurden für die verschiedenen Konfigurationen Richtwerte der minimal und durchschnittlich erzeugten Energie als Funktion der Bahnparameter bestimmt. Dabei sind diese Richtwerte auf prograden Orbits näherungsweise nur Funktionen der Bahnhöhe und der Inklination und auf SSOs der Bahnhöhe und der LTDN. Die Standardkonfiguration erzeugt die maximale Energie auf prograden Orbits bei hohen Inklinationen und großen Bahnhöhen und auf SSOs auf Dämmerungsorts. Die Konfigurationen mit ausklappbaren Paneelen erzeugen hingegen mehr Energie auf niedrig inklinierten Orbits, auf denen die Sonnenlichtperioden generell kürzer sind, wobei die Verläufe stärkere Gradienten aufweisen. Diese Konfigurationen sind insbesondere auf SSOs mit einer LTDN von 12 h oder 24 h zu bevorzugen.

Im Vergleich erzeugen die Konfigurationen mit ausklappbaren Paneelen im Mittel auf prograden Orbits mit Ausnahme von polnahen Orbits stets mehr Energie als die Standardkonfiguration. Im schlechtesten Fall ist dies jedoch nur auf Umlaufbahnen mit Inklinationen unter ca.  $30^\circ$  der Fall, auf höher inklinierten Bahnen treten stets auch längere Sonnenlichtperioden auf, während der die ausgeklappten Paneele von Nachteil sind. Auf SSOs ist die Standardkonfigurationen bei einer LTDN zwischen 3,5 h und 8,5 h sowie zwischen 15,5 h und 20 h zu bevorzugen. Mit Hilfe der gegebenen Energiewerte kann für verschiedene Umlaufzeiten, also verschiedene Bahnhöhen, die entsprechende maximal vom Satellitensystem aufnehmbare Leistung abgeschätzt werden.

Die mögliche Bodenkontaktzeit steigt ebenfalls mit zunehmender Bahnhöhe. Da die beiden betrachteten Bodenstationen bei ca.  $53^\circ$  (Bremen) bzw.  $68^\circ$  Nord (Inuvik in Kanada) liegen, kann auf Orbits mit Inklinationen unter ca.  $40^\circ$  kein ausreichender Kontakt zu einer Bodenstation hergestellt werden. Die optimale Inklination für eine maximale Bodenkontaktzeit ist dabei etwas größer als die nördliche Breite der Bodenstation. Die Bodenstation in Bremen kann im optimalen Fall zu einem Satelliten in 650 km Höhe über 50 Minuten pro Tag Kontakt herstellen, die Bodenstation in Inuvik maximal 70 Minuten pro Tag.

## 11.1 Ausblick

Mit den in dieser Arbeit dargestellten Ergebnissen lassen sich für zukünftige CLAVIS-Missionen Richtwerte für die Sonnenlichtzeiten, die erzeugte Energie, die damit maximal vom Satellitensystem aufnehmbare Leistung sowie die Bodenkontaktzeiten

zu den beteiligten Bodenstationen abschätzen. Für die von den Solarpaneelen erzeugte Energie lassen sich für verschiedene Konfigurationen mit unterschiedlich vielen Solarzellen pro Paneel Richtwerte als Funktion der Bahnparameter (Bahnhöhe, Inklination, LTDN) oder genauere Werte als Funktion der Bahnhöhe und der Sonnenlichtdauer näherungsweise bestimmen. Damit kann das Energieversorgungssystem zukünftiger CLAVIS-Missionen grob ausgelegt werden. Darüber hinaus können der Einfluss der Sonnenstrahlung auf den Thermalhaushalt sowie die möglichen Bodenkontaktzeiten abgeschätzt werden. Die Ergebnisse eignen sich jedoch nicht für eine detaillierte Auslegung einer Mission oder des Energieversorgungssystems. Dazu sind im Einzelfall genauere Analysen nötig.

Sollen bei zukünftigen CLAVIS-Missionen andere Solarpaneelstellungen, Ausklappwinkel oder ein anderes Lageregelungskonzept verwendet werden, müssen die im Anhang und auf der DVD zur Arbeit wiedergegebenen und dokumentierten MATLAB-Skripte entsprechend angepasst werden. Allgemein stellt sich beim Vorentwurf eines Raumfahrzeugs oft die Frage, wie die Solarpaneele bei gegebenem Orbit und gegebener Lage am besten angeordnet werden sollten. Für eine mögliche entsprechende Arbeit, die generelle Empfehlungen für diese Problematik entwickelt, können die hier verwendeten Ansätze als Orientierung dienen.

Darüber hinaus wäre auch eine Verifikation der für die AISat-Mission gegebenen Ergebnisse anhand an Bord des Satelliten gemessenen Werten interessant, insbesondere im Hinblick auf den Wirkungsgrad der Solarzellen sowie den Einfluss der Taumelbewegungen.

# Literaturverzeichnis

- [1] ANSELMO, PORTELLI, CROWTHER, TREMAYNE-SMITH, ALBY, BACCINI, BONNAL, ALWES, FLURY, JEHN, KLINKRAD, *European Code of Conduct for Space Debris Mitigation*, ASI, BNSC, CNES, DLR, ESA, 2004, online verfügbar:  
<http://www.cnsa.gov.cn/n615708/n676979/n676983/n893604/appendix/...2008529151013.pdf>  
letzter Zugriff: 18.04.2011
- [2] CORNELIUS S., *Entwurf eines Entfaltungsmechanismus für Solarpaneele auf CLAVIS*, Diplomarbeit, Institut für Luft- und Raumfahrtsysteme, TU Braunschweig, 2011
- [3] MESSERSCHMID, E., BERTRAND, R., POHLEMANN, F., *Raumstationen - Systeme und Nutzung*, Springer, 1997
- [4] BENDISCH J., *Raumfahrttechnik 2 - Raumfahrtmissionen*, Vorlesungsskript, Institut für Luft- und Raumfahrtsysteme, TU Braunschweig, 2008
- [5] STANDISH E., *Keplerian Elements for Approximate Positions of the Major Planets*, NASA JPL/Caltech, online verfügbar:  
[http://ssd.jpl.nasa.gov/txt/aprx\\_pos\\_planets.pdf](http://ssd.jpl.nasa.gov/txt/aprx_pos_planets.pdf)  
letzter Zugriff: 18.04.2011
- [6] VALLADO, D., *Fundamentals of Astrodynamics and Applications*, 3rd edition, Microcosm Press/Springer, 2007
- [7] STEINER, W., SCHAGERL, M., *Raumflugmechanik - Dynamik und Steuerung von Raumfahrzeugen*, Springer, 2004
- [8] STRAIN, A., *User Manual: Deployed Electronic Power System For Part #: DEPS2-0096-CS*, ClydeSpace User Manual, Document No.: C3D-USM-5023-DLR-EPS2, Issue A, 2010
- [9] MCLAREN, V., *User Manual: Standalone 20Wh Battery*, ClydeSpace User Manual, Document No.: C3-USM-5026-CS-BAT-20Wh, Issue B, 2010
- [10] DROBCZYK, M., *CLAVIS Powerbudget*, Internes Dokument des DLR Instituts für Raumfahrtsysteme, 2011
- [11] LEY, W., WITTMANN, K., HALLMANN, W., *Handbuch der Raumfahrttechnik*, 3. Auflage, Hanser, 2008

- [12] BALENZATEGUI, J., CHENLO, F., *Measurement and analysis of angular response of bare and encapsulated silicon solar cells*, Solar Energy Materials and Solar Cells, Volume 86, pp. 53-83, 2005
- [13] PISACANE, V., *Fundamentals of Space Systems*, Oxford University Press, 2nd edition, 2005
- [14] -, *28.3% Ultra Triple Junction (UTJ) Solar Cells*, Spectrolab Datenblatt, 2010, online verfügbar:  
<http://www.spectrolab.com/DataSheets/cells/PV%20UTJ%20Cell%205-20-10.pdf>  
letzter Zugriff: 18.04.2011
- [15] -, *Space Environment Information System*, ESA online tool:  
<http://www.spenvis.oma.be/>  
letzter Zugriff: 18.04.2011
- [16] WERTZ, J., LARSON, W., *Space Mission Analysis and Design*, 3rd edition, Microcosm Press/Springer, 1999
- [17] -, *Lithium Ion Polymer Cell*, ClydeSpace Material Safety Data Sheet
- [18] DROBCZYK, M., *CLAVIS Linkbudget*, Internes Dokument des DLR Instituts für Raumfahrtssysteme, 2011
- [19] GRIFFIN, M., FRENCH, J., *Space Vehicle Design*, 2nd edition, AIAA, 2004
- [20] WALTER, U., *Astronautics*, WILEY-VCH, 2008
- [21] -, *STK 9.2.1 Hilfe*, AGI, 2010
- [22] HEIDECKER, A., *Private Unterhaltung am DLR Institut für Raumfahrtssysteme*
- [23] MCCAMISH, S., *Distributed Autonomous Control of multiple Spacecraft during close Proximity Operations*, Dissertation, Naval Postgraduate School, 2007, online verfügbar:  
<http://handle.dtic.mil/100.2/ADA475928>  
letzter Zugriff: 18.04.2011
- [24] -, *STK 9.2.1 Object Model Diagram*, AGI, 2010, im Installationsverzeichnis von STK verfügbar:  
... \AGI\STK 9\Help\ObjectModel\pdf\ObjectModel\_diagram.pdf
- [25] VALLADO, D., *MATLAB-Funktionen zu [6]*, online verfügbar:  
<http://www.microcosminc.com/Vallado>  
letzter Zugriff: 18.04.2011
- [26] -, *MATLAB 7.9.0 Hilfe*, The MathWorks Inc., 2009
- [27] DELGADO, M., *Radiation Pressure - Modeling the Space Environment*, Vorlesungspräsentation, Universidad Politécnica de Madrid, 2008, online verfügbar:  
<http://ocw.upm.es/ingenieria-aeroespacial/modeling-the-space-environment/...>  
...contenidos/material-de-clase/mse01\_intro\_ocw.pdf  
letzter Zugriff: 18.04.2011

- [28] SCHITTKOWSKI, K., *Data Fitting and Experimental Design in Dynamical Systems with EasyFit*, Department of Computer Science, University of Bayreuth, 2008, online verfügbar:  
[http://www.ai7.uni-bayreuth.de/easy\\_fit.pdf](http://www.ai7.uni-bayreuth.de/easy_fit.pdf)  
letzter Zugriff: 18.04.2011
- [29] PHILLIPS, J., *zunzun.com*, frei verfügbares Tool zur Bestimmung von Näherungsgleichungen, online verfügbar:  
<http://zunzun.com>  
letzter Zugriff: 18.04.2011
- [30] SPRÖWITZ, T., BAUER, W., DROBCZYK, M., NOHKA, F., HEIDECKER, A., *CLAVIS - Erste Schritte zu einer standardisierten NanoSat-Plattform*, Internationale Luft- und Raumfahrttausstellung Berlin, 2010

# Abbildungsverzeichnis

2.1	CLAVIS in der Standardkonfiguration . . . . .	12
2.2	Schematische Darstellung der Konfigurationen mit ausklappbaren Paneelen . . . . .	13
2.3	CLAVIS in der AISat-Konfiguration . . . . .	13
3.1	Kernschatten und Halbschatten der Erde, nicht maßstabsgetreu . . . . .	16
3.2	Geometrie zur näherungsweisen Berechnung der maximalen Eklipsendauer für $\beta = 0^\circ$ . . . . .	18
3.3	Maximale Eklipsendauer als Funktion der Bahnhöhe für kreisförmige Orbits . . . . .	19
3.4	Geometrie zur Berechnung des Grenzfalles des Betawinkels, bei dem der Orbit komplett in der Sonne liegt . . . . .	19
3.5	Geometrie des Erdorbits zur Bestimmung von $\theta$ mit der Position der Erde am 21.3. zur Definition des Frühlingspunktes, die Exzentrizität des Erdorbits ist stark übertrieben (nicht maßstabsgetreu) . . . . .	23
3.6	Verläufe von Betawinkel und Eklipsendauer für einen Orbit mit $i = 50^\circ$ und $h = 500$ km über ein Jahr mit dem Startzeitpunkt 01.01.2011 0 h unter Berücksichtigung der $J_2$ -Störung . . . . .	25
3.7	Verläufe von Betawinkel und Eklipsendauer für einen Orbit mit $i = 10^\circ$ und $h = 500$ km über ein Jahr mit dem Startzeitpunkt 01.01.2011 0 h unter Berücksichtigung der $J_2$ -Störung . . . . .	26
3.8	Verläufe von Betawinkel und Eklipsendauer für einen Orbit mit $i = 80^\circ$ und $h = 500$ km über ein Jahr mit dem Startzeitpunkt 01.01.2011 0 h unter Berücksichtigung der $J_2$ -Störung . . . . .	27
3.9	Verläufe von Betawinkel und Eklipsendauer für einen SSO mit $h = 500$ km und $LTDN = 6$ h über ein Jahr mit dem Startzeitpunkt 01.01.2011 0 h unter Berücksichtigung der $J_2$ -Störung . . . . .	29
3.10	Verläufe von Betawinkel und Eklipsendauer für einen SSO mit $h = 500$ km und $LTDN = 7:30$ h über ein Jahr mit dem Startzeitpunkt 01.01.2011 0 h unter Berücksichtigung der $J_2$ -Störung . . . . .	29
4.1	Solarpaneele von CLAVIS aus [8] . . . . .	31
4.2	Strom-Spannungs-Kennlinie einer Solarzelle . . . . .	32
5.1	Geometrie der Funkverbindung zwischen einem Satelliten in einem kreisförmigen, erdnahen Orbit und einer Bodenstation . . . . .	44
8.1	Grafische Darstellung des Modells der AISat-Konfiguration in STK . . . . .	66

8.2	Vergleich der Berechnung der bereitgestellten Leistung der Solarpaneele der Standardkonfiguration über eine Sonnenlichtphase in einem SSO in 650 km Höhe mit einer LTDN von 10 h . . . . .	69
9.1	SSO der AISat-Mission Anfang März aus Richtung der Sonne betrachtet	71
9.2	Verläufe des Einstrahlwinkels auf die verschiedenen Solarpaneele der AISat-Konfiguration von CLAVIS über einen Orbit . . . . .	71
9.3	Verläufe der erzeugten Leistung der verschiedenen Solarpaneele der AISat-Konfiguration von CLAVIS über einen Orbit . . . . .	72
9.4	Verlauf der Ladung der Batterie der AISat-Konfiguration von CLAVIS über einen Orbit bei konstanter Leistungsaufnahme, die senkrechten Linien deuten den Beginn und das Ende der Sonnenlichtperiode an . . . .	73
9.5	Verlauf der gesamten erzeugten Leistungen der verschiedenen Konfigurationen von CLAVIS über einen Umlauf, auf einem Orbit wie bei der AISat-Mission Anfang März . . . . .	75
9.6	Verlauf der Dauer der Sonnenlichtperioden während der AISat-Mission, Vergleich zwischen dem $J_2$ -Propagator und dem HPOP . . . . .	77
9.7	Verläufe der über eine Sonnenlichtperiode erzeugten Energie des oberen Paneels, der seitlichen Paneele und der Gesamtenergie als Funktion der Missiondauer für die AISat-Mission mit exakter Nadirausrichtung, Vergleich zwischen dem $J_2$ -Propagator und dem HPOP . . . . .	78
9.8	Verläufe der über eine Sonnenlichtperiode erzeugten Energie des oberen Paneels, der seitlichen Paneele und der Gesamtenergie als Funktion der Missiondauer für die AISat-Mission, Vergleich zwischen exakter Nadirausrichtung und Taumelbewegungen verschieden großer Amplitude .	80
9.9	Verläufe der Marge zwischen erzeugter und benötigter Energie als Funktion der Missionsdauer für die AISat-Mission, Vergleich zwischen exakter Nadirausrichtung und Taumelbewegungen verschieden großer Amplitude . . . . .	82
9.10	Verlauf der DoD der Batterie als Funktion der Missionsdauer für die AISat-Mission mit exakter Nadirausrichtung, Vergleich zwischen dem $J_2$ -Propagator und dem HPOP . . . . .	83
9.11	Verläufe der Aufladerate der Batterie als Funktion der Missionsdauer für die AISat-Mission mit exakter Nadirausrichtung, Vergleich zwischen dem $J_2$ -Propagator und dem HPOP . . . . .	83
9.12	Verläufe der über eine Sonnenlichtperiode erzeugten Energie des oberen Paneels, der seitlichen Paneele und der Gesamtenergie als Funktion der Missiondauer für die AISat-Mission, Vergleich zwischen exakter Nadir- und Zenitausrichtung . . . . .	85
9.13	Verlauf der vom Satelliten in Zenitausrichtung maximal aufnehmbaren Leistung, so dass die Batterie in jeder Sonnenlichtphase wieder voll geladen wird . . . . .	86
9.14	Kumulierte Bodenkontaktzeiten und kumulierte Bodenkontaktzeiten mit ausreichendem CNR für die AISat-Mission in Abhängigkeit der Missionzeit . . . . .	87

10.1	Verläufe der durchschnittlichen Sonnenlichtdauer als Funktion der Inklination auf einer Bahnhöhe von 650 km auf Grundlage der Simulationen verschiedener Startzeitpunkte und -rektaszensionen und der in Abschnitt 10.2.2 gegebenen Näherungsgleichung . . . . .	91
10.2	Verläufe von kürzester, mittlerer und längster Sonnenlichtdauer als Funktion von Bahnhöhe und Inklination, Mittelwerte über alle Startzeitpunkte und -rektaszensionen, siehe auch Abbildung G.1 . . . . .	93
10.3	Verläufe der längsten Sonnenlichtperioden für polare Orbits als Funktion der RAAN für verschiedene Bahnhöhen . . . . .	95
10.4	Verläufe der kürzesten und längsten Sonnenlichtperioden für SSOs als Funktion der LTDN und der Bahnhöhe, siehe auch Abbildung G.2 . . . . .	97
10.5	Verläufe der Sonnenlichtperioden durchschnittlicher Dauer für SSOs als Funktion der Bahnhöhe und der LTDN, Simulationsergebnisse als Punkte und angenäherter Verlauf, siehe auch Abbildung G.2 . . . . .	99
10.6	Streudiagramm der von der Standardkonfiguration erzeugten Energie als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen . . . . .	103
10.7	Verlauf der erzeugten Energie aller seitlichen Paneele der Standardkonfiguration als Funktion der Dauer der Sonnenlichtperiode und der Bahnhöhe, Simulationsergebnisse als Punkte und Verlauf nach der Näherungsgleichung (10.8), siehe auch Abbildung G.3 . . . . .	104
10.8	Verlauf der erzeugten Energie des oberen Paneels als Funktion der Dauer der Sonnenlichtperiode und der Bahnhöhe, Simulationsergebnisse als Punkte und Verlauf nach der Näherungsgleichung (10.8), siehe auch Abbildung G.4 . . . . .	105
10.9	Streudiagramm der von der Konfiguration mit um 60° ausgeklappten Paneelen erzeugten Gesamtenergie als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen . . . . .	106
10.10	Verlauf der erzeugten Energie aller seitlichen Paneele der Konfiguration mit um 60° ausgeklappten Paneelen als Funktion der Dauer der Sonnenlichtperiode und der Bahnhöhe, Simulationsergebnisse als Punkte und Verlauf nach der Näherungsgleichung (10.8), siehe auch Abbildung G.5 . . . . .	107
10.11	Streudiagramm der von der Konfiguration mit um 90° ausgeklappten Paneelen erzeugten Gesamtenergie als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen . . . . .	108
10.12	Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.6 . . . . .	110
10.13	Minimalwerte der durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugten Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.7 . . . . .	112
10.14	Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der LTDN, siehe auch Abbildung G.8 . . . . .	113



10.15	Durchschnittlich durch die seitlichen Paneele der Standardkonfiguration erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, Simulationsergebnisse als Punkte und angenäherter Verlauf, siehe auch Abbildung G.8 . . . . .	115
10.16	Minimale durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.9 . . . . .	116
10.17	Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um 60° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.10 . . . . .	118
10.18	Durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um 60° ausgeklappten Paneelen insgesamt erzeugte Energie bei minimaler Gesamtenergie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.11 . . .	119
10.19	Durchschnittlich durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.12 . . . . .	121
10.20	Minimal durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.13 . . . . .	122
10.21	Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um 60° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der Inklination, siehe auch Abbildung G.14 . . . . .	123
10.22	Durchschnittlich durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Bahnhöhe und der LTDN für SSOs, siehe auch Abbildung G.16 . . . . .	126
10.23	Vergleich der durch die Standardkonfiguration, die Konfiguration mit um 90° ausgeklappten Paneelen und die Konfiguration mit um 30° verkippten Paneelen erzeugten Energie über eine Sonnenlichtperiode im Oktober auf SSOs verschiedener LTDN in 650 km Höhe . . . . .	127
10.24	Von dem Satellitensystem bei vollständiger Aufladung der Batterie aufnehmbare Leistung als Funktion der erzeugten Energie über eine Sonnenlichtperiode für verschiedene Bahnhöhen . . . . .	128
10.25	Durchschnittlich mögliche Bodenkontaktdauer mit der Bodenstation in Bremen pro Tag als Funktion der Inklination für verschiedene Bahnhöhen	130
10.26	Vergleich der Geometrie des möglichen Bodenkontakts zwischen der Bodenstation in Bremen und Satelliten auf Orbits in 200 km und 650 km Höhe bei einer Inklination von 60° . . . . .	131
10.27	Durchschnittlich mögliche Bodenkontaktdauer mit der Bodenstation in Inuvik pro Tag als Funktion der Inklination für verschiedene Bahnhöhen	132

10.28	Verlauf der Bahnhöhe von CLAVIS über ein Jahr unter Berücksichtigung der Störeinflüsse für Missionen auf verschiedenen Bahnhöhen jeweils bei hoher, niedriger und mittlerer Sonnenaktivität . . . . .	133
E.1	Vereinfachtes Struktogramm eines MATLAB-Skripts zur Berechnung der Rohdaten zur Analyse der AISat-Mission . . . . .	192
F.1	Vereinfachtes Struktogramm eines MATLAB-Skripts zur Berechnung der Rohdaten zur Analyse der zukünftigen CLAVIS-Missionen . . . . .	208
G.1	Verläufe von kürzester, mittlerer und längster Sonnenlichtdauer als Funktion der Inklination für verschiedene Bahnhöhen, Mittelwerte über alle Startzeitpunkte und -rektaszensionen, zu Abbildung 10.2 . . . . .	233
G.2	Verläufe von kürzester, mittlerer und längster Sonnenlichtdauer für SSOs als Funktion der LTDN für verschiedene Bahnhöhen, zu den Abbildungen 10.4 und 10.5 . . . . .	234
G.3	Verläufe der erzeugten Energie aller seitlichen Paneele der Standardkonfiguration als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen, zu Abbildung 10.7 . . . . .	234
G.4	Verlauf der erzeugten Energie des oberen Paneels als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen, zu Abbildung 10.8 . . . . .	235
G.5	Verlauf der erzeugten Energie aller seitlichen Paneele der Konfiguration mit um 60° ausgeklappten Paneelen als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen, zu Abbildung 10.10 . . . . .	235
G.6	Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.12 . . . . .	236
G.7	Minimalwerte der durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugten Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.13 . . . . .	236
G.8	Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der LTDN für verschiedene Bahnhöhen, zu den Abbildungen 10.14 und 10.15 . . . . .	237
G.9	Minimale durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.16 . . . . .	237
G.10	Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um 60° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.17 . . . . .	238

G.11	Durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um 60° ausgeklappten Paneelen insgesamt erzeugte Energie bei minimaler Gesamtenergie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.18 . . .	238
G.12	Durchschnittlich durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.19 . . . . .	239
G.13	Minimal durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.20 . . . . .	239
G.14	Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um 60° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.21 . . . . .	240
G.15	Im schlechtesten Fall durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um 60° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen . . . . .	240
G.16	Durchschnittlich durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der LTDN für SSOs für verschiedene Bahnhöhen, zu Abbildung 10.22 . . . . .	241
G.17	Minimal durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der LTDN für SSOs für verschiedene Bahnhöhen . . . . .	241
H.1	Übersicht über die zur Abschätzung der kürzesten, längsten und durchschnittlichen Sonnenlichtzeiten aus den Orbitparametern zu verwendenden Gleichungen (Gl.), Abbildungen (Abb.) und Tabellen (Tab.) . . .	242
H.2	Übersicht über die zur Abschätzung der erzeugten Energie aus der Dauer der Sonnenlichtzeiten zu verwendenden Gleichungen (Gl.), Abbildungen (Abb.) und Tabellen (Tab.) . . . . .	242
H.3	Übersicht über die zur Abschätzung der erzeugten Energie aus den Orbitparametern zu verwendenden Gleichungen (Gl.), Abbildungen (Abb.) und Tabellen (Tab.) . . . . .	243
I.1	Work Breakdown Structure . . . . .	244
I.2	Projektplanung als Gantt-Diagramm . . . . .	261

# Tabellenverzeichnis

4.1	Betriebsmodi der AISat-Konfiguration mit Leistungsaufnahme und Auslastungen der Subsysteme . . . . .	35
10.1	Faktoren der Näherungsgleichung (10.1) zur Abschätzung der minimalen Sonnenlichtdauer als Funktion von Bahnhöhe und Inklination für Inklinationen zwischen $0^\circ$ und $90^\circ$ . . . . .	93
10.2	Faktoren der Näherungsgleichung (10.1) zur Abschätzung der durchschnittlichen Sonnenlichtdauer als Funktion von Bahnhöhe und Inklination für Inklinationen zwischen $0^\circ$ und $90^\circ$ . . . . .	94
10.3	Faktoren der Näherungsgleichung (10.3) zur Abschätzung der maximalen Sonnenlichtdauer als Funktion von Bahnhöhe und Inklination für Inklinationen zwischen $0^\circ$ und $i_{\text{grenz}}$ . . . . .	95
10.4	Faktoren der Näherungsgleichung (10.4) zur Abschätzung der minimalen Sonnenlichtdauer als Funktion von Bahnhöhe und LTDN für SSOs .	98
10.5	Faktoren der Näherungsgleichung (10.4) zur Abschätzung der maximalen Sonnenlichtdauer als Funktion von Bahnhöhe und LTDN für SSOs, auf denen keine Umläufe ohne Eklipse auftreten . . . . .	98
10.6	Faktoren der Näherungsgleichung (10.4) zur Abschätzung der durchschnittlichen Sonnenlichtdauer als Funktion von Bahnhöhe und LTDN für SSOs für Sonnenlichtdauern unter $t_{\text{ill,grenz}}$ . . . . .	100
10.7	Faktoren der Näherungsgleichung (10.8) zur Abschätzung der von den seitlichen Paneelen der Standardkonfiguration erzeugten Energie als Funktion von Sonnenlichtdauer in Minuten und Bahnhöhe in Kilometern . . . . .	104
10.8	Faktoren der Näherungsgleichung (10.8) zur Abschätzung der vom oberen Paneel erzeugten Energie als Funktion von Sonnenlichtdauer in Minuten und Bahnhöhe in Kilometern . . . . .	105
10.9	Faktoren der Näherungsgleichung (10.8) zur Abschätzung der von den seitlichen Paneelen der Konfiguration mit um $60^\circ$ ausgeklappten Paneelen erzeugten Energie als Funktion von Sonnenlichtdauer in Minuten und Bahnhöhe in Kilometern . . . . .	108
10.10	Faktoren der Näherungsgleichung (10.9) zur Abschätzung der von den seitlichen Paneelen und des oberen Paneels der Standardkonfiguration durchschnittlich erzeugten Energie als Funktion der Bahnhöhe und der Inklination . . . . .	111

10.11	Faktoren der Näherungsgleichung (10.9) zur Abschätzung der von den seitlichen Paneelen und des oberen Panels der Standardkonfiguration minimal erzeugten Energie als Funktion der Bahnhöhe und der Inklinat	112
10.12	Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch die seitlichen Paneele der Standardkonfiguration durchschnittlich erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien unter $E_{grenz,ave,side}^{max}$	114
10.13	Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch das obere Paneel durchschnittlich erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien über $E_{grenz,ave,top}^{min}$	115
10.14	Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch die seitlichen Paneele der Standardkonfiguration minimal erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien unter $E_{grenz,min,side}^{max}$	117
10.15	Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch das obere Paneel der Standardkonfiguration bei minimaler Gesamtenergie erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien unter $E_{grenz,min,top}^{min}$	117
10.16	Faktoren der Näherungsgleichung (10.15) zur Abschätzung der von den seitlichen Paneelen der Konfiguration mit um 60° ausgeklappten Paneelen durchschnittlich erzeugten Energie als Funktion von Bahnhöhe und Inklinat	118
10.17	Faktoren der Näherungsgleichung (10.16) zur Abschätzung der von den seitlichen Paneelen und des oberen Panels der Konfiguration mit um 60° ausgeklappten Paneelen bei minimaler Gesamtenergie erzeugten Energie als Funktion der Bahnhöhe und der Inklinat	120
10.18	Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch die seitlichen Paneele der Konfiguration mit um 60° ausgeklappten Paneelen bei durchschnittlicher Gesamtenergie erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien über $E_{grenz,ave,side}^{min}$	124
10.19	Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch die seitlichen Paneele der Konfiguration mit um 60° ausgeklappten Paneelen bei minimaler Gesamtenergie erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für Energien über $E_{grenz,min,side}^{min}$ und außerhalb der Bereiche der LTDN zwischen 5 h und 7 h sowie zwischen 11 h und 13 h	125
10.20	Faktoren der Näherungsgleichung (10.10) zur Abschätzung der durch das obere Paneel der Konfiguration mit um 60° ausgeklappten Paneelen bei minimaler Gesamtenergie erzeugten Energie als Funktion der Bahnhöhe und der LTDN für SSOs für erzeugte Energien über $E_{grenz,min,top}^{min}$	125
B.1	Eingaben in das SPENVIS-Tool in der Kategorie <i>Coordinate Generators</i>	158
B.2	Eingaben in das SPENVIS-Tool in der Kategorie <i>Radiation sources and effects</i> , Unterkategorie <i>Radiation sources</i>	159
B.3	Eingaben in das SPENVIS-Tool in der Kategorie <i>Radiation sources and effects</i> , Unterkategorie <i>Solar Cell radiation damage</i>	159

# Anhang A

## Grundlegende MATLAB-Skripte

In diesem Abschnitt sind die Quellcodes einiger grundlegender MATLAB-Funktionen wiedergegeben, die in verschiedenen weiteren MATLAB-Skripten verwendet werden. Darüber hinaus werden die Skripte wiedergegeben, die zur Erstellung der in Kapitel 3 gezeigten Diagramme des Verlaufs von Betawinkel und Eklipsendauer über der Zeit verwendet wurden.

### A.1 Häufig verwendete MATLAB-Funktionen

#### jday.m [25]

Zweck: Berechnung des Julianischen Datums aus gegebenem Datum (Jahr, Monat, Tag, Stunde, Minute, Sekunde)

```
% -----
%
%
%                               function jday.m
%
% this function finds the julian date given the year, month, day, and time.
% author      : david vallado              719-573-2600   27 may 2002
% revisions   :
%
% inputs      description                      range / units
% year        - year                          1900 .. 2100
% mon         - month                        1 .. 12
% day         - day                          1 .. 28,29,30,31
% hr          - universal time hour           0 .. 23
% min         - universal time min            0 .. 59
% sec         - universal time sec           0.0 .. 59.999
% whichtype   - julian .or. gregorian calender  'j' .or. 'g'
% outputs     :
% jd          - julian date                  days from 4713 bc
% locals      :
% none.
% coupling    :
% none.
% references   :
% vallado     2007, 189, alg 14, ex 3-14
% jd = jday(yr, mon, day, hr, min, sec)
% -----
function jd = jday(yr, mon, day, hr, min, sec)

% ----- implementation -----
jd = 367.0 * yr ...
    - floor( (7 * (yr + floor( (mon + 9) / 12.0) ) ) * 0.25 ) ...
    + floor( 275 * mon / 9.0 ) ...
    + day + 1721013.5 ...
    + ( (sec/60.0 + min) / 60.0 + hr ) / 24.0;
% - 0.5 * sign(100.0 * yr + mon - 190002.5) + 0.5;
```

**m2phi.m**

Zweck: Numerische Berechnung der wahren Anomalie aus der mittleren Anomalie mit Hilfe der Exzentrizität

```
function [phi] = m2phi(M,e)

% m2phi: calculate true anomaly from mean anomaly and eccentricity
%   input: mean anomaly M [deg], eccentricity e [1]
%   output: true anomaly phi [deg]

% M in rad
M = M*pi/180;

% use Newton's method to calculate eccentric anomaly numerically
found = false;
i = 1;
EA = M;
while found == false
    g = EA-e*sin(EA)-M;
    g_deriv = 1-e*cos(EA);
    EA = EA-(g/g_deriv);
    % abort criterion #1 -> value found
    if abs(g/g_deriv) < 0.000001
        found = true;
    end
    i = i+1;
    % abort criterion #2 -> no value found after 100 iterations
    if i > 100
        display('Warning: calculation of ecc anomaly did not converge!');
        found = true;
    end
end
% calculate true anomaly from eccentric anomaly
trAn = 2*atan(tan(EA/2)*sqrt(abs((1+e)/(1-e))));
if trAn < 0
    trAn = trAn+2*pi;
end
phi = trAn*180/pi;
end
```

**start\_LTDN.m**

Zweck: Berechnung der LTDN aus gegebener RAAN und Julianischem Datum

```
function [LTDN] = start_LTDN(OM,JD)

% calculate LTDN from start date and RAAN
% input:
% OM: RAAN [deg]
% JD: start date as julian date, between 01/2000 and 2050
% output:
% LTDN [h]

RAAN = OM*pi/180;
juldat = JD;

% days since epoch J2000.0
juldat_diff = juldat-2451545;

% centurys since epoch
juldat_cty = juldat_diff/36525;

% orbital parameters of Earth orbit at epoch and rate of change of elements
% per century
% eccentricity
e_0=0.001671123;
e_dot=-0.00004392;
% mean Longitude
L_0=100.46457166*pi/180;
L_dot=35999.37244981*pi/180;
% Longitude of perihelion = argument of perihelion, as RAAN=0
om_0=102.93768193*pi/180;
om_dot=0.32327364*pi/180;

% orbital parameters of Earth at start date
e = e_0+e_dot*juldat_cty;
L = L_0+L_dot*juldat_cty;
om = om_0+om_dot*juldat_cty;

% mean anomaly of Earth at start date in degrees
M = (L-om)*180/pi;

% true anomaly
```

```

phi = m2phi(M,e);

% longitude of Earth in radians, interval [0:2pi]
long_e = (phi*pi/180)+om;
if long_e > 2*pi && long_e <= 4*pi
    long_e = long_e-2*pi;
elseif long_e > 4*pi
    long_e = long_e-4*pi;
end

% Local Time of first point of Aries at start date in rad
LT_FP = 2*pi-long_e;

% Local Time of Ascending Node of start orbit = LT_FP + OM [rad]
LTAN_rad = LT_FP+RAAN;
if LTAN_rad > 2*pi && LTAN_rad <= 4*pi
    LTAN_rad = LTAN_rad-2*pi;
elseif LTAN_rad > 4*pi
    LTAN_rad = LTAN_rad-4*pi;
end

% in degrees
LTAN_deg = LTAN_rad*180/pi;
% in hours
LTAN      = LTAN_deg*24/360;
% LTDN
if LTAN >= 12
    LTDN = LTAN-12;
else
    LTDN = LTAN+12;
end
end

```

### solar\_power.m

Zweck: Berechnung der pro Flächeneinheit von der Sonne eingestrahltten Energie aus Start- und Endzeitpunkt einer Sonnenlichtperiode nach Gleichung (4.2)

```

function [power] = solar_power(start_time,stop_time,epoch)

% function to calculate solar power per area as function of the time of the
% year.
% inputs:
% start_time: start time of lighting interval in Epoch Minutes
% stop_time:  stop time of lighting interval in Epoch Minutes
% epoch:      day of year of epoch
% output:
% power:      solar power per unit area in [W/m^2]

epdays_ave = (stop_time+start_time)/(2*60*24);
dayofyear   = epoch+epdays_ave;
if dayofyear > 366
    dayofyear = dayofyear-365; % no 0th day, 365.99 -> 1.01
end
day_aft_3jan = dayofyear-3;
if day_aft_3jan < 0
    day_aft_3jan = day_aft_3jan+365;
end
power = 1371*(1+0.033*cos(2*pi*day_aft_3jan/365));
end

```

### UTCG2EpMin.m

Zweck: Umrechnung eines Datums als Zeichenkette im UTCG-Format in die Zeit nach einer gegebenen Epoche in Minuten

```

function [EpMin] = UTCG2EpMin (epoch_start,time)
% convert UTCG date format:
% 01 Jan 2000 00:00:00.000000000 as string
% to Epoch Minutes after Epoch_start in UTCG string format

% save DD MMM YYYY HH:MM:SS.SSSSSSSSS as numbers
day      = str2double(time(1:2));
day_start = str2double(epoch_start(1:2));
month_str = time(4:6);
month_start_str = epoch_start(4:6);
switch(month_str)
    case('Jan')
        month=1;
    case('Feb')

```



```

        month=2;
    case('Mar')
        month=3;
    case('Apr')
        month=4;
    case('May')
        month=5;
    case('Jun')
        month=6;
    case('Jul')
        month=7;
    case('Aug')
        month=8;
    case('Sep')
        month=9;
    case('Oct')
        month=10;
    case('Nov')
        month=11;
    case('Dec')
        month=12;
end
switch(month_start_str)
    case('Jan')
        month_start=1;
    case('Feb')
        month_start=2;
    case('Mar')
        month_start=3;
    case('Apr')
        month_start=4;
    case('May')
        month_start=5;
    case('Jun')
        month_start=6;
    case('Jul')
        month_start=7;
    case('Aug')
        month_start=8;
    case('Sep')
        month_start=9;
    case('Oct')
        month_start=10;
    case('Nov')
        month_start=11;
    case('Dec')
        month_start=12;
end
year      = str2double(time(8:11));
year_start = str2double(epoch_start(8:11));
hours     = str2double(time(13:14));
hours_start = str2double(epoch_start(13:14));
minutes   = str2double(time(16:17));
minutes_start = str2double(epoch_start(16:17));
seconds   = str2double(time(19:30));
seconds_start = str2double(epoch_start(19:30));

% calculate Julian Date of epoch and time via jday function by Vallado
JD_time = jday(year,month,day,hours,minutes,seconds);
JD_start = jday(year_start,month_start,day_start,hours_start,minutes_start,seconds_start);

% calculate time difference in days
EpDays = JD_time-JD_start;
EpMin  = EpDays*24*60;
end

```

## A.2 MATLAB-Skripte zu den Abbildungen in Kapitel 3

### beta\_angle\_eclipse\_duration\_vs\_time.m

Zweck: Berechnen der theoretischen Verläufe von Betawinkel und Eklipsendauer über die Missionszeit von einem Jahr und ausgeben eines Diagramms (hier beispielhaft für Abbildung 3.6)

```

clear all
close all
clc

% start date:
YYYY = 2011;

```

```

MM    = 01;
DD    = 01;
hh    = 00;
mm    = 00;

% circular orbit: e=0, om=0
a_0   = 6878.14;
i_0   = 50*pi/180;
OM_0  = 0*pi/180;

mu = 398600;
% orbital period in seconds
T_U = 2*pi*sqrt(a_0^3/mu);
% number of orbits per year
num_orb_year = 365.25*24*3600/T_U;

% Julian Date from start date
juldat = jday(YYYY,MM,DD,hh,mm,0.0);

% time step vector after start date, 1 entry per hour
t_pre = 0:1/24:365;
t      = t_pre;

theta = zeros(length(t),1);
OM     = zeros(length(t),1);
beta  = zeros(length(t),1);
t_E   = zeros(length(t),1);

% loop over all time steps
for i = 1:length(t);
    % calculate angles and eclipse duration for every time step
    theta(i) = angle_sun_vernal_equinox(juldat,t(i));
    OM(i)    = RAAN_J2(OM_0,i_0,a_0,t(i));
    beta(i)  = beta_angle(theta(i),OM(i),i_0);
    t_E(i)   = eclipse_time(a_0,beta(i));
end

t_E_ave = zeros(length(t_E),1);
for i = 1:length(t_E)
    t_E_ave(i) = mean(t_E);
end

figure
[AX,H1,H2] = plotyy(t,beta*180/pi,t,t_E/60);
set(H1,'LineWidth',2,'Color',[0 0 0])
set(H2,'LineStyle','--','LineWidth',2,'Color',[0 0 1])
set(AX(1),'YColor',[0 0 0],'FontName','Palatino','FontSize',12)
set(AX(2),'YColor',[0 0 1],'FontName','Palatino','FontSize',12)
ylim([-100 100])
set(AX(2),'Ylim',[0 50])
xlabel('Zeit [d]')
ylabel('Betawinkel [°]')
set(get(AX(2),'Ylabel'),'String','Eklipsendauer [min]','FontName','Palatino','FontSize',12)
legend('Betawinkel','Eklipsendauer','Location','NorthEast')
grid on

```

In diesem Skript werden insgesamt vier Funktionen verwendet, die nicht standardmäßig in MATLAB implementiert sind. Diese werden im Folgenden wiedergegeben.

### angle\_sun\_vernal\_equinox.m

Zweck: Berechnung des Winkels  $\theta$  zwischen Sonne und Frühlingspunkt nach Gleichung (3.8)

```

function [theta]= angle_sun_vernal_equinox (juldat_start,t)

% calculation of angle theta (angle between sun and first point of Aries in
% the ecliptic) at time t [days=24h] after start date juldat_start [Julian
% Date], output of theta [rad]

% days since epoch epoch J2000.0
juldat_diff=(juldat_start+t)-2451545;

% centuries since epoch
juldat_cty=juldat_diff/36525;

% orbital parameters of Earth @ J2000.0 and rates of change
% eccentricity
e_0   = 0.001671123;
e_dot = -0.00004392;
% mean Longitude
L_0   = 100.46457166*pi/180;
L_dot = 35999.37244981*pi/180;

```

```

% Longitude of perihelion = argument of perihelion, as RAAN=0
om_0 = 102.93768193*pi/180;
om_dot = 0.32327364*pi/180;

% orbital parameters of Earth at start date
e = e_0+e_dot*juldat_cty;
L = L_0+L_dot*juldat_cty;
om = om_0+om_dot*juldat_cty;

% mean anomaly of Earth at start date in deg
M = (L-om)*180/pi;

% true anomaly from mean anomaly and eccentricity
phi = m2phi(M,e);

% theta=180°+phi+om
theta_0 = pi+phi*pi/180+om;

% get theta in interval [0...2*pi]
if theta_0 > 2*pi && theta_0 < 4*pi
    theta = theta_0-2*pi;
elseif theta_0 > 4*pi
    theta = theta_0-4*pi;
else
    theta = theta_0;
end
end

```

## RAAN\_J2.m

Zweck: Berechnung der aufgrund der  $J_2$ -Störung veränderten RAAN zu einem bestimmten Zeitpunkt bei gegebener Inklination und großer Halbachse

```

function [OM]=RAAN_J2(OM_0,i,a,t)

% calculation of RAAN OM [rad] for given point of time t [days=24h] after
% start date using inclination i [rad], semimajor axis a [km] and RAAN at
% start date OM_0 [rad]

% rate of change of RAAN due to J_2-perturbation
OM_dot = (-2.06474E14*a^(-7/2)*cos(i))*pi/180;

% RAAN at time t
OM_pre = OM_0+OM_dot*t;

% get RAAN in interval [0...2*pi]
if OM_pre >= 0
    num = floor(OM_pre/(2*pi));
    OM = OM_pre-num*2*pi;
else
    num = ceil(abs(OM_pre)/(2*pi));
    OM = OM_pre+num*2*pi;
end

end

```

## beta\_angle.m

Zweck: Berechnung des Betawinkels aus dem Winkel zwischen Sonne und Frühlingspunkt, RAAN und Inklination nach Gleichung (3.9)

```

function [beta]=beta_angle(theta,OM,i)

% calculation of beta angle [rad] from angle between sun and first point of
% Aries in the ecliptic theta [rad], RAAN OM [rad],
% inclination i [rad]

sin_beta = cos(theta)*sin(OM)*sin(i)-sin(theta)*cos(23.5*pi/180)*cos(OM)*...
    sin(i)+sin(theta)*sin(23.5*pi/180)*cos(i);

beta = asin(sin_beta);

```

**eclipse\_time.m**

Zweck: Berechnung der Eklipsendauer aus Betawinkel und großer Halbachse mit Hilfe der Gleichungen (3.1) und (3.6)

```
function [t_E]=eclipse_time(a,beta)

% calculation of theoretical eclipse duration t_E [s] as function of
% semimajor axis a [km] and beta angle [rad]

% Earth radius
R_0 = 6378;

% gravitational constant of Earth [km^3/s^2]:
mu = 398600;

% half of shadow angle in rad
if ((R_0/a)^2-(sin(beta))^2) >= 0
    alpha = asin(sqrt((R_0/a)^2-(sin(beta))^2)/cos(beta));
    % orbital period in seconds
    T_U = 2*pi*sqrt(a^3/mu);
    % eclipse duration in seconds
    t_E = T_U*2*alpha/(2*pi);
else
    t_E = 0;
end
end
```

# Anhang B

## SPENVIS Eingaben

In diesem Kapitel wird eine Übersicht der Eingabeparameter für das von der ESA online zur Verfügung gestellte Tool SPENVIS [15] gegeben. Dieses Tool berechnet für einen gegebenen Typ Solarzellen die der Strahlungsbelastung auf einem bestimmten Orbit äquivalente Fluenz von 1 MeV-Elektronen (siehe auch Abschnitt 4.3.2). Die Eingabeparameter sind nach den Kategorien von SPENVIS aufgeteilt. Alle hier nicht aufgeführten Parameter sind unerheblich.

Kategorie: Coordinate Generators

Unterkategorie: Spacecraft trajectories

Trajectory generation	use orbit generator
Number of mission segments	1
Mission end	total mission duration
Mission duration	1.0 years
Account for solar radiation pressure	no
Account for atmospheric drag	no
Orbit type	heliosynchronous
Orbit start	calender date
	01 Sep 2011 00:00:00
Representative:	
Trajectory duration [days]	1
Altitude [km]	650
Local time of ascending node [hr]	22

**Tabelle B.1:** Eingaben in das SPENVIS-Tool in der Kategorie *Coordinate Generators*

Kategorie: Radiation sources and effects

Unterkategorie: Radiation sources

Trapped proton and electron fluxes:	
Proton model	AP-8 (Solar Maximum)
Electron Model	AE-8 (Solar Maximum)
Trapped proton flux anisotropy:	
Trapped proton anisotropy model	Badhwar & Konradi 1990 MAX
Long-term solar particle fluences:	
Solar particle model	Rosenqvist et al. (2005, 2007)
Confidence Level	95 %
Magnetic shielding	default
Galactic cosmic ray fluxes:	
Ion range	H to Ni
GCR model at 1 AU	ISO 15390
	ISO-15390 standard model
Solar activity data	mission epoch
Magnetic shielding	default

**Tabelle B.2:** Eingaben in das SPENVIS-Tool in der Kategorie *Radiation sources and effects*, Unterkategorie *Radiation sources*

Kategorie: Radiation sources and effects

Unterkategorie: Solar Cell radiation damage

Damage equivalent fluences for solar cells:	
Cell type	multiple junction
Manufacturer	Spectrolab 3J
Coverglass thickness	125 $\mu\text{m}$

**Tabelle B.3:** Eingaben in das SPENVIS-Tool in der Kategorie *Radiation sources and effects*, Unterkategorie *Solar Cell radiation damage*

# Anhang C

## Steuerung von STK mit MATLAB

In diesem Abschnitt wird zunächst erläutert, wie die Verbindung zwischen STK und MATLAB hergestellt und die hier verwendeten Objekte (siehe Kapitel 7) in MATLAB definiert werden. Anschließend wird das Erstellen und Auslesen von Datentabellen aus STK mit MATLAB beschrieben. Dabei wird hier davon ausgegangen, dass dem Leser die MATLAB-Programmierungsumgebung und die wichtigsten Befehle bekannt sind. Daher wird hier vornehmlich auf die Schnittstelle zwischen STK und MATLAB eingegangen.

### C.1 Herstellen der Verbindung zwischen STK und MATLAB

Es gibt grundsätzlich zwei Möglichkeiten, eine Verbindung zwischen STK und MATLAB aufzubauen. Die direktere Methode ist die Verwendung des „Connect“-Moduls von STK [21]. Dieses Modul beinhaltet zum einen einen Grundstock an häufig gebrauchten Anweisungen (Erstellen eines neuen Objekts, Einstellen des Lageprofils eines Satelliten, etc.), die als *mexConnect*-Befehle zusammengefasst werden. Diese Befehle beginnen jeweils mit *stk...* und besitzen als Argumente eine definierte Menge an Parametern, die zur Ausführung des Befehls in STK benötigt werden. Darüber hinaus gibt es einen Katalog mit einer Vielzahl von weiteren Anweisungen, mit denen alle möglichen Einstellungen vorgenommen und Reports ausgegeben werden können. Diese Anweisungen werden mit dem Befehl *stkExec* aufgerufen, der als Argument eine Zeichenkette mit allen relevanten Parametern erhält. Diese Befehle sind für die Anwendung mit MATLAB definiert worden und können daher ohne weiteres verwendet werden.

Darüber hinaus besteht aber auch die Möglichkeit, das sogenannte „Object Model“ zu verwenden, das als Teil des „Software Development Kits“ in STK enthalten ist. Die darin enthaltenen Funktionen müssen nicht über MATLAB aufgerufen werden, sondern können mit praktisch jeder Programmiersprache verwendet werden. In der entsprechenden Dokumentation in STK [21] finden sich dazu stets Beispiele in Visual Basic, C# und C++. Der Vorteil dieses Pakets ist, dass mit den darin enthaltenen Befehlen eine Vielzahl von Attributen und Methoden jedes denkbaren STK-Objekts angesprochen werden kann. Eine Übersicht hierzu findet sich in [24]. Damit bestehen umfassendere Möglichkeiten, die Funktionalität von STK bis auf die Ebene einzelner

Objekte zu nutzen. Es ist außerdem im Rahmen des Object Models möglich, die Connect-Befehle uneingeschränkt zu verwenden.

Im Laufe dieser Arbeit stellte sich heraus, dass die Connect-Befehle einige Schwächen bei der Ausgabe von benutzerdefinierten Reports besitzen. Allerdings sind sie unerlässlich bei der Verwendung des Solar Panel Tools (siehe Abschnitt 7.1.2) und erleichtern die Eingabe von Parametern der Sender- und Empfängerobjekte. Daher wird hier STK über das Object Model gesteuert, so dass eine größere Auswahl an Befehlen zur Verfügung steht.

Damit STK mit MATLAB-Skripten gesteuert werden kann, wird hier stets vorausgesetzt, dass die STK-Anwendung läuft, aber kein Szenario geladen ist. Die Verbindung zu STK wird hergestellt mit den folgenden Befehlszeilen.

```
uiApp    = actxGetRunningServer('STK9.application');  
root     = uiApp.Personality2;
```

Darin wird zunächst das Objekt *uiApp* definiert, das die interne Verbindung zur laufenden STK-Anwendung beinhaltet. In der zweiten Zeile wird dann das Stammobjekt *root* mit Hilfe des vorherigen Objekts definiert. Dies ist das Objekt der höchsten Ebene und dient als Stammobjekt für alle weiteren zu definierenden Objekte, auch für das Szenarioobjekt. Auf dieses Objekt wird zurückgegriffen, wenn Connect-Befehle verwendet werden sollen und um globale Einstellungen, wie zum Beispiel die zu verwendenden Maßeinheiten, zu definieren (siehe Abschnitt C.2.1).

Anhand der beiden gegebenen Codezeilen erkennt man ein grundsätzliches Prinzip der Verwendung des STK Object Models mit MATLAB. Es existieren verschiedene Befehle, mit denen alle möglichen Objekte erstellt werden können. Der Zugriff auf Attribute und Methoden der Objekte erfolgt dann durch den Punkt-Operator.

## C.2 Erstellen des STK-Szenarios mit MATLAB

Alle zur Simulation der verschiedenen Szenarios verwendeten MATLAB-Skripte beinhalten zu Beginn die Definition des entsprechenden Szenarios und der benötigten Objekte. Diese Definitionen unterscheiden sich in den verschiedenen Skripten nur in wenigen Punkten, weshalb sie hier allgemeingültig dargestellt werden. Um die einzelnen Befehle zu verstehen, muss die Objektstruktur im Object Model von STK jedoch genauer bekannt sein. Bisher wurde der Einfachheit halber stets von Attributen und Methoden eines Objekts gesprochen. Streng genommen ist aber ein Objekt in STK (wie zum Beispiel ein Satellit oder das Stammobjekt) eine Klasse und besitzt mehrere Schnittstellen (sogenannte *Interfaces*), in denen Attribute und Methoden definiert sind. Dabei gibt es auch hierarchische Strukturen innerhalb der Schnittstellen einer Klasse [24], so dass ein Aufruf einer bestimmten Methode über verschiedene Ebenen dieser Hierarchie erfolgen kann. Der Zugriff auf eine untergeordnete Schnittstelle erfolgt dabei jeweils mit Punkt-Operator.



## C.2.1 Definition der Maßeinheiten

Um mögliche Unklarheiten über die verwendeten grundlegenden Einheiten zu vermeiden, werden sie zunächst definiert. Die folgenden Codezeilen definieren die Datumsangabe, die Entfernung, die Zeiteinheit sowie die Einheit für Winkel.

```
root.UnitPreferences.SetCurrentUnit('DateFormat', 'UTCG');
root.UnitPreferences.SetCurrentUnit('DistanceUnit', 'm');
root.UnitPreferences.SetCurrentUnit('TimeUnit', 'sec');
root.UnitPreferences.SetCurrentUnit('AngleUnit', 'deg');
```

Wie bereits erwähnt, sind die Maßeinheiten Attribute des Stammobjekts und werden über die Methode *SetCurrentUnit* der Schnittstelle *UnitPreferences* definiert. Diese Methode bekommt als Eingabe eine Zeichenkette mit der zu definierenden Größe und eine mit der entsprechenden Einheit. Bei der Datumsangabe steht *UTCG* für ein Gregorianisches Datum in Koordinierter Weltzeit der Form '01 Jan 2000 00:00:00.000'. Diese Zeitangabe wird für den Aufruf des Solar Panel Tools benötigt und wird deshalb hier verwendet. In Skripten, in denen das Tool nicht verwendet wird, ist diese Art der Datumsangabe jedoch eher unpraktisch, da sie immer eine Zeichenkette fordert und keine numerischen Werte enthält. In solchen Fällen werden im Rahmen dieser Arbeit auch die Einheiten *EpMin* für die Minuten nach dem Startzeitpunkt des Szenarios und *JDate* für die Angabe als Julianisches Datum verwendet (siehe auch Abschnitt 3.3.3).

## C.2.2 Definition des Szenarioobjekts

Das Szenarioobjekt wird in MATLAB durch die folgenden Codezeilen definiert.

```
root.NewScenario('CLAVIS_Matlab_neu');
Scenario = root.GetObjectFromPath('*/');
scen_start = '15 Aug 2011 00:00:00.000000000';
scen_stop = '15 Aug 2012 00:00:00.000000000';
Scenario.StopTime = scen_stop;
Scenario.StartTime = scen_start;
Scenario.TerrainCollection.Item(0).UseTerrain = 1;
```

Darin wird zunächst ein neues Szenario mit der Methode *NewScenario* erstellt, die als Eingabe den Namen des Szenarios erhält. Danach wird ein Objekt mit dem Namen *Scenario* erstellt, das für zukünftige Zugriffe zur Verfügung steht. Dazu wird die Methode *GetObjectFromPath* verwendet, die als Eingabe einen Pfad erhält. Der hier verwendete Pfad entspricht dem Szenarioobjekt, weitere untergeordnete Objekte werden dabei wie eine Ordnerstruktur behandelt (siehe auch Abschnitt C.2.6).

In den folgenden Zeilen werden der Anfangs- und Endzeitpunkt des Szenarios als Gregorianisches Datum als Variablen definiert. Daraufhin werden diese Variablen den entsprechenden Attributen des Szenarios zugewiesen. Hier dargestellt ist die Definition für eine Datumsangabe als Zeichenketten im Gregorianischen Datum. Um aus einer solchen Datumsangabe das Julianische Datum zu berechnen, kann die Funktion *jday* verwendet werden, die Jahr, Monat, Tag, Stunde, Minute und Sekunde als Parameter übergeben bekommt und daraus das Julianische Datum berechnet [6] (siehe Abschnitt A.1). In der letzten Zeile wird angegeben, dass die in STK enthaltene Datei mit den Daten zur Höhe der Erdoberfläche über dem Referenzellipsoid verwendet werden soll (siehe auch Abschnitt 7.1.1). Dazu wird das Attribut *UseTerrain* des entsprechenden

Eintrags (*Item*) in der Liste (Schnittstelle *TerrainCollection*) auf eins gesetzt.

### C.2.3 Manipulation der Fenster

Nachdem ein Szenarioobjekt erstellt wurde, öffnet STK automatisch zwei Fenster zur Visualisierung, ein 2D-Fenster für die Subspur des Satelliten und ein 3D-Fenster für die Gestalt des Orbits. Beide Fenster werden hier nicht weiter benötigt und erhöhen die Rechenzeit, da sie ständig aktualisiert werden. Deshalb werden sie mit den folgenden Zeilen wieder geschlossen.

```
root.ExecuteCommand('Window3D * Remove WindowID 1');
root.ExecuteCommand('Window2D * Remove WindowID 1');
```

Dabei werden über die *ExecuteCommand*-Methode des Stammobjekts Connect-Befehle (*Window3D* und *Window2D*) zur Manipulation der Fenster aufgerufen. Die Methode erhält als Eingabe eine Zeichenkette, die aus dem Namen des Connect-Befehls, dem Pfad des betroffenen Objekts (hier das Sternchen für das Szenario), einem Schlüsselwort zur Art der Manipulation des Fensters und den für diese Manipulation notwendigen Parametern, in diesem Fall die Identifikationsnummer des Fensters, besteht.

Darüber hinaus wird auch das Hauptfenster der STK-Anwendung durch die folgenden Codezeilen manipuliert.

```
root.ExecuteCommand('Application / Maximize');
root.ExecuteCommand('Application / Hide');
```

Auch hier wird ein Connect-Befehl verwendet, mit dem das STK-Hauptfenster zunächst maximiert und dann versteckt wird. Dazu enthält das Argument neben dem Schlüsselwort *Application* den Pfad der Anwendung, nur ein Schrägstrich, und den entsprechenden Befehl. Die Maximierung erhöht wie in Abschnitt 7.1.2 dargelegt, die Genauigkeit des Solar Panel Tools. Das Fenster wird zusätzlich versteckt, damit keine automatischen Aktualisierungen vorgenommen werden, die die Rechenzeit erhöhen.

### C.2.4 Definition des Satellitenobjekts

Mit den folgenden MATLAB-Codezeilen wird ein neues Satellitenobjekt erstellt und dessen Orbit zugewiesen und propagiert.

```
Satellite = Scenario.Children.New('eSatellite','CLAVIS');
Satellite.SetPropagatorType('ePropagatorJ2Perturbation');
Satellite.Propagator.UseScenarioAnalysisTime;
Satellite.Propagator.InitialState.Representation.AssignClassical ...
... ('eCoordinateSystemJ2000',7028140,0.0,97.9908,0.0,293.06,0.0);
Satellite.Propagator.Propagate();
```

Zunächst wird das Objekt *Satellite* mit Hilfe der *New*-Methode der *Children*-Schnittstelle des Szenarioobjekts erstellt, wobei der Typ und der Name des Objekts als Argumente übergeben werden. Daraufhin wird dem Satellitenobjekt über die *Set-PropagatorType*-Methode ein Propagator zugewiesen, in diesem Fall der einfache *J<sub>2</sub>*-Propagator. Über die *Propagator*-Schnittstelle des Satellitenobjekts wird dann der

Zeitraum der Propagation dem Zeitraum des Szenarios gleichgesetzt. Außerdem wird dem Satelliten mit der Methode *AssignClassical* ein Orbit zugewiesen, der über ein Koordinatensystem und die KEPLERSchen Bahnelemente in der Reihenfolge große Halbachse, Exzentrizität, Inklination, Argument des Perigäums, RAAN und wahre Anomalie definiert wird. Abschließend wird der Orbit durch Aufruf der Methode *Propagate* propagiert.

Die Zeile, in der der zu propagierende Orbit definiert wird, ist hier aufgrund ihrer Länge umgebrochen dargestellt. In MATLAB bedeuten drei Punkte am Ende einer Zeile, dass der Befehl in der nächsten Zeile fortgeführt wird. Die drei Punkte in der folgenden Zeile sind dabei kein MATLAB-Code und dienen hier nur der besseren Lesbarkeit des Codes. Soll diese Befehlszeilen verwendet werden, müssten diese gelöscht werden. Dabei muss beachtet werden, dass dieser Umbruch nicht innerhalb einer Zeichenkette verwendet werden kann. Da aber bei der Verwendung der Schnittstelle zwischen MATLAB und STK oft lange Zeichenketten auftreten, sind im Folgenden auch solche Zeilen auf analoge Weise umgebrochen wiedergegeben. Dies dient lediglich der übersichtlicheren Darstellung und ist kein gültiger MATLAB-Code.

Im Anschluss wird die Lage des Satelliten durch die folgenden Befehlszeilen definiert.

```
Satellite.SetAttitudeType('eAttitudeStandard');
Satellite.Attitude.Basic.SetProfileType('eProfileSpinAboutNadir');
Satellite.Attitude.Basic.Profile.Rate = 0;
Satellite.Attitude.Basic.Profile.Offset = 0;
Satellite.Attitude.Basic.Profile.Epoch = '15 Aug 2011 00:00:00.0';
```

Darin wird zunächst eingestellt, dass ein vordefiniertes, „Standard“-Lageprofil verwendet werden soll. In der zweiten Zeile wird dann über die *Attitude*-Schnittstelle des Satellitenobjekts das in Abschnitt 7.1.2 angesprochene Lageprofil ausgewählt, das die z-Achse des Satelliten in Richtung Nadir ausrichtet. Daraufhin werden die Drehrate, der Offset und die Startzeit dieses Profils definiert. Letztere stimmt dabei im Allgemeinen mit der Startzeit des Szenarios überein. Soll eine externe Datei zur Definition der Lage verwendet werden, ist die folgende Zeile ausreichend, in der der *Load*-Methode der Pfad der externen Datei übergeben wird.

```
Satellite.Attitude.External.Load('G:\STK\CLAVIS.a');
```

Darüber hinaus wird dem Satelliten über die *VO*-Schnittstelle des Objekts das 3D-Modell zugewiesen, was mit den folgenden Zeile erreicht wird.

```
Satellite.VO.Model.ModelType = 'eModelFile';
Satellite.VO.Model.ModelData.FileName = 'STKData\VO\Models\Space\CLAVIS.mdl';
```

Darin wird zuerst definiert, dass eine Modelldatei (\*.mdl) verwendet werden soll und daraufhin der Pfad der Datei dem Attribut *Filename* zugewiesen. Hierbei ist der Unterschied zu der Verwendung der externen Datei bei der Definition der Lage des Satelliten zu beachten.

Abschließend müssen noch die Körper definiert werden, die für die Bestimmung der Schattenzeiten herangezogen werden sollen. Wie in Abschnitt 3.2 dargestellt, soll hier

nur die Erde verwendet werden. Da aber in der Standardeinstellung auch der Mond berücksichtigt wird, wird dies in den folgenden Codezeilen geändert.

```
if Satellite.EclipseBodies.IsCentralBodyAssigned('Moon') == 1
    Satellite.EclipseBodies.UseCustomizedList = 1;
    Satellite.EclipseBodies.AssignCentralBody('Earth');
end
```

Darin wird zunächst abgefragt, ob der Mond in der Liste der *EclipseBodies* vorkommt. Ist dies der Fall, wird eine benutzerdefinierte Liste verwendet, die dann zunächst keine Einträge besitzt. Dieser Liste wird dann nur die Erde hinzugefügt.

## C.2.5 Definition des Bodenstationsobjekts

Zur Definition des Bodenstationsobjekt wird der folgende Code verwendet.

```
Bremen = Scenario.Children.New('eFacility','Bremen');
Bremen.Position.AssignGeodetic(53.111, 8.862, 0);
Bremen.UseTerrain = 1;

Elev_Constraint_Bre = ...
    ... Bremen.AccessConstraints.AddConstraint('eCstrElevationAngle');
Elev_Constraint_Bre.EnableMin = 1;
Elev_Constraint_Bre.Min      = 10;
```

In der ersten Zeile wird ähnlich dem Satellitenobjekt (siehe Abschnitt C.2.4) das Bodenstationsobjekt *Bremen* erstellt. Diesem Objekt wird über die *Position*-Schnittstelle die Position in geodätischen Koordinaten zugewiesen, wobei die Höhe auf Null belassen wird. Dazu wird in der folgenden Zeile die Option gesetzt, dass die Datei mit den Höhendaten der Erdoberfläche verwendet werden soll (siehe Abschnitt 7.1.3). Danach wird der Elevationsbereich, in dem Satelliten von der Bodenstation aus sichtbar sind, begrenzt. Dazu wird das *Elev\_Constraint\_Bre*-Objekt mit der *AddConstraint*-Methode erstellt, der die Art der Beschränkung als Argument übergeben wird. Daraufhin wird die Beschränkung durch einen Minimalwert gewählt. Dieser wird dann auf 10° gesetzt (siehe Abschnitt 5.2).

## C.2.6 Definition des Senderobjekts

Das Senderobjekt mit dem Namen *Downlink* wird als ein dem Satellitenobjekt untergeordnetes Objekt definiert, wobei der Befehl dem der Definition des Satellitenobjekts ähnelt. Daraufhin werden die in Abschnitt 5.3 erläuterten Parameter gesetzt. Beides wird mit den folgenden Befehlszeilen erreicht.

```
Transmitter = Satellite.Children.New('eTransmitter','Downlink');

root.ExecuteCommand('Transmitter */Satellite/CLAVIS/Transmitter/Downlink ...
    ... SetValue Model Simple_Transmitter_Model');

root.ExecuteCommand('Transmitter */Satellite/CLAVIS/Transmitter/Downlink ...
    ... SetValue Model.Frequency 436e+6');

root.ExecuteCommand('Transmitter */Satellite/CLAVIS/Transmitter/Downlink ...
    ... SetValue Model.EIRP 1.621810097');

root.ExecuteCommand('Transmitter */Satellite/CLAVIS/Transmitter/Downlink ...
    ... SetValue Model.Modulator.AutoScaleBandwidth false');
```

```

root.ExecuteCommand('Transmitter */Satellite/CLAVIS/Transmitter/Downlink ...
... SetValue Model.Modulator.Bandwidth 10000');

root.ExecuteCommand('Transmitter */Satellite/CLAVIS/Transmitter/Downlink ...
... AddItem Model.PostTransmitGainsLosses.GainLossList');
root.ExecuteCommand('Transmitter */Satellite/CLAVIS/Transmitter/Downlink ...
... SetValue ...
... Model.PostTransmitGainsLosses.GainLossList[0].Identifier PointLoss');
root.ExecuteCommand('Transmitter */Satellite/CLAVIS/Transmitter/Downlink ...
... SetValue ...
... Model.PostTransmitGainsLosses.GainLossList[0].Gain 0.050118723');

```

Die Definition der charakteristischen Parameter des Senders erfolgt hier über den Connect-Befehl *Transmitter*. Dieser bekommt zunächst den Pfad des betroffenen Objekts als Eingabe, wobei hier von dem Szenario ausgehend das Satelliten- und das Senderobjekt berücksichtigt werden. Darauf folgt bei einer Reihe von Werten das Schlüsselwort *SetValue* und anschließend die zu definierende Eigenschaft, wobei auch hier eine Hierarchie der verschiedenen Parameter berücksichtigt werden muss. Schließlich folgt der eigentliche Wert, der dem Attribut zugewiesen werden soll. Um zusätzliche Verluste hinzufügen zu können (siehe Abschnitt 7.1.4) muss der entsprechenden Liste zunächst ein Eintrag hinzugefügt werden. Daraufhin wird dieser Eintrag benannt, in diesem Falle zu *PointLoss* und schließlich auf einen bestimmten Wert gesetzt. Dabei muss beachtet werden, dass alle Werte hier in Standardeinheiten (Hz, W) oder dimensionslos angegeben werden und nicht in Dezibel.

## C.2.7 Definition des Empfängerobjekts

Das Empfängerobjekt wird grundsätzlich analog zum Senderobjekt definiert, wobei hier die Frequenz und die Bandbreite automatisch an die des Senderobjekts angepasst werden. Daher müssen nur die Zahlenwerte für das Verhältnis  $\frac{G_r}{T_{sys}}$  des Empfängers (siehe Abschnitt 5.3.3) sowie für die verschiedenen Verluste auf der Funkstrecke und im Empfänger gesetzt werden. Dies wird durch die folgenden Codezeilen erreicht.

```

Receiver = Bremen.Children.New('eReceiver','Receiver');

root.ExecuteCommand('Receiver */Facility/Bremen/Receiver/Receiver ...
... SetValue Model.Simple_Receiver_Model');

root.ExecuteCommand('Receiver */Facility/Bremen/Receiver/Receiver ...
... SetValue Model.GOverT 0.08416642');

root.ExecuteCommand('Receiver */Facility/Bremen/Receiver/Receiver ...
... AddItem Model.PreReceiveGainsLosses.GainLossList');
root.ExecuteCommand('Receiver */Facility/Bremen/Receiver/Receiver ...
... SetValue ...
... Model.PreReceiveGainsLosses.GainLossList[0].Identifier GSPointLoss');
root.ExecuteCommand('Receiver */Facility/Bremen/Receiver/Receiver ...
... SetValue ...
... Model.PreReceiveGainsLosses.GainLossList[0].Gain 0.630957344');

root.ExecuteCommand('Receiver */Facility/Bremen/Receiver/Receiver ...
... AddItem Model.PreReceiveGainsLosses.GainLossList');
root.ExecuteCommand('Receiver */Facility/Bremen/Receiver/Receiver ...
... SetValue ...
... Model.PreReceiveGainsLosses.GainLossList[1].Identifier GSLineLosses');
root.ExecuteCommand('Receiver */Facility/Bremen/Receiver/Receiver ...
... SetValue ...
... Model.PreReceiveGainsLosses.GainLossList[1].Gain 0.501187233');

```

```

root.ExecuteCommand( 'Receiver */Facility/Bremen/Receiver/Receiver ...
... AddItem Model.PreReceiveGainsLosses.GainLossList');
root.ExecuteCommand( 'Receiver */Facility/Bremen/Receiver/Receiver ...
... SetValue ...
... Model.PreReceiveGainsLosses.GainLossList[2].Identifier PolarisLoss');
root.ExecuteCommand( 'Receiver */Facility/Bremen/Receiver/Receiver ...
... SetValue ...
... Model.PreReceiveGainsLosses.GainLossList[2].Gain 0.891250938');

root.ExecuteCommand( 'Receiver */Facility/Bremen/Receiver/Receiver ...
... AddItem Model.PreReceiveGainsLosses.GainLossList');
root.ExecuteCommand( 'Receiver */Facility/Bremen/Receiver/Receiver ...
... SetValue ...
... Model.PreReceiveGainsLosses.GainLossList[3].Identifier IonosLoss');
root.ExecuteCommand( 'Receiver */Facility/Bremen/Receiver/Receiver ...
... SetValue ...
... Model.PreReceiveGainsLosses.GainLossList[3].Gain 0.891250938');

```

Darin wird zunächst das Empfängerobjekt als untergeordnetes Objekt der Bodenstation Bremen erstellt und das Verhältnis von Antennengewinn zu Systemrauschtemperatur definiert. Daraufhin werden vier verschiedene Verluste der entsprechenden Liste hinzugefügt. Dazu wird wie beim Senderobjekt zunächst ein neuer Eintrag in der Liste erstellt, dem dann eine Bezeichnung und ein Wert zugewiesen wird. Somit können die Verluste im Empfänger (*GSPointLoss* und *GSLineLoss*) und auf der Funkstrecke (*PolarisLoss* und *IonosLoss*) berücksichtigt werden.

## C.3 Erstellen von STK-Reports mit MATLAB

Die in STK berechneten Daten werden wie in Abschnitt 7.1.1 bereits erwähnt in sogenannten Reports ausgegeben. Diese Reports können auch in MATLAB definiert und die enthaltenen Daten in Objekten gespeichert und damit weiterverarbeitet werden. In den folgenden Abschnitten werden die dafür nötigen Schritte für verschiedene Fälle erläutert.

### C.3.1 Sonnenlichtzeiten und Energieerzeugung

Beispielhaft zeigen die folgenden Zeilen, wie die Sonnenlichtzeiten („Lighting Times“) für einen Satelliten mit STK bestimmt und in MATLAB gespeichert werden. Die Speicherung der Daten der von den Solarpaneelen erzeugten Energie bzw. des Vektors zur Sonne wird analog dazu durchgeführt.

```

DataProv = ...
... Satellite.DataProviders.Item('Lighting Times').Group.Item('Sunlight');

RptElems = {'Start Time'; 'Stop Time'; 'Duration'};

Results = ...
... DataProv.ExecElements(Scenario.StartTime, Scenario.StopTime, RptElems);

Data = Results.DataSets;

```

Darin wird zunächst das Objekt *DataProv* erstellt, in dem die Art der auszugebenen Daten definiert wird. Dazu wird in der Schnittstelle *DataProviders* des Satellitenobjekts die entsprechende Kategorie ausgewählt; in diesem Fall die Zeitabschnitte in denen der Satellit beleuchtet wird. Einige dieser Kategorien enthalten noch Unterkategorien,

die mit dann mit Hilfe des Schlüsselwortes *Group* ausgewählt werden können. In dem hier dargestellten Fall wird das Sonnenlicht ausgewählt, eine andere mögliche Option wäre die Beleuchtung durch den Mond. Die Kategorien und Unterkategorien der „Data Providers“ können im sogenannten „Report & Graph Manager“ des STK GUI eingesehen werden.

In der folgenden Zeile wird der Vektor der gewünschten Ausgabedaten definiert. Die Bezeichnungen müssen dabei natürlich mit in der gewählten Kategorie vorhandenen übereinstimmen. Darauf folgt die eigentliche Berechnung der Daten, die im Objekt *Results* gespeichert werden. Dieses wird unter Verwendung der *ExecElements*-Methode des *DataProv*-Objekts erstellt. Diese Methode bekommt den Start- und Endzeitpunkt des Szenarios sowie den soeben definierten Vektor der auszugebenen Elemente als Parameter übergeben. Die Zeitpunkte müssen dabei in dem aktuellen Datumsformat vorliegen (siehe Abschnitt C.2.1) und die entsprechenden Ausgaben werden ebenfalls in diesem Format bereitgestellt. Schließlich werden die im *Results*-Objekt enthaltenen Daten über die Methode *DataSets* in dem neuen Objekt *Data* gespeichert.

Wie bereits erwähnt gelten die hier dargestellten Möglichkeiten, die gewünschten Daten zu den Sonnenlichtzeiten zu erhalten, auch für die beleuchtete Fläche der Solarpaneele und anderer für die erzeugte Energie relevanter Werte. Dabei muss natürlich beachtet werden, dass das Solar Panel Tool ausgeführt werden muss, bevor auf die entsprechenden Daten zurückgegriffen werden kann. Bei der Berechnung von Bodenkontaktzeiten muss allerdings ein zusätzlicher Schritt beachtet werden, wie im folgenden Abschnitt gezeigt wird.

### C.3.2 Bodenkontaktzeiten und Übertragungsqualität

Die folgenden Codezeilen zeigen die Erzeugung von Reportdaten zu Bodenkontaktzeiten und zum Linkbudget, in diesem Fall die Berechnung der CNR zu jedem Zeitpunkt während aller Kontaktzeiten.

```
RptElems_Acc = {'Start Time'; 'Stop Time'; 'Duration'};
RptElems_LiB = {'C/N'};

Access = Receiver.GetAccessToObject(Transmitter);
Access.AccessTimePeriod = 'eObjectAccessTime';
Access.ComputeAccess;

DataProvider_Acc = Access.DataProviders.Item('Access Data');
DataProvider_LiB = Access.DataProviders.Item('Link Information');

Results_Acc      = DataProvider_Acc.ExecElements ...
    ... (Scenario.StartTime, Scenario.StopTime, RptElems_Acc);
Results_LiB      = DataProvider_LiB.ExecElements ...
    ... (Scenario.StartTime, Scenario.StopTime, time_step, RptElems_LiB);

Data_Acc         = Results_Acc.DataSets;
Data_LiB         = Results_LiB.DataSets;
```

Zunächst werden darin analog zu dem oben beschriebenen Fall für die Sonnenlichtzeiten die Vektoren mit den gewünschten Ausgabedaten definiert, wobei hier zwei verschiedene Vektoren auftreten. Zum einen interessieren die Startzeitpunkte,

Endzeitpunkte und Dauern aller Bodenkontaktzeiten, zum anderen die CNR, hier als *C/N* bezeichnet. Um diese Werte berechnen zu lassen wird ein *Access*-Objekt über die *GetAccessToObject*-Methode des Empfängerobjekts erstellt, die als Parameter das Senderobjekt übergeben bekommt. Diesem Objekt wird in der folgenden Zeile als Zeitspanne zur Berechnung der Zeitraum zugewiesen, während dem beide beteiligten Objekte gültig sind. Die eigentliche Berechnung der Daten erfolgt dann über die *ComputeAccess*-Methode des *Access*-Objekts.

Anschließend werden wiederum analog zu den Darstellungen in Abschnitt C.3.1 die *Data Providers* für Kontaktzeiten und Linkbudget getrennt definiert, da sich die Daten in unterschiedlichen Kategorien befinden. Im Gegensatz zu den Sonnenlichtzeiten gibt es dabei keine Unterkategorien. Daraufhin werden die Daten wiederum in *Results*-Objekten gespeichert, wobei ein wichtiger Unterschied zwischen der Berechnung der Kontaktzeiten und der CNR auffällt.

Die Kontaktzeiten sind hier, ähnlich wie die Sonnenlichtzeiten oben, unabhängig von den diskreten Zeitschritten der Simulation, da hier Anfangszeitpunkt, Endzeitpunkt und Dauer unabhängig davon ausgegeben werden. Die CNR soll hingegen zu jedem Zeitpunkt einer Bodenkontaktphase ausgegeben werden. Daher muss nun der Methode *ExecElements* das Zeitintervall als zusätzlicher Parameter übergeben werden. Dieser wird nach dem Endzeitpunkt des Szenarios und vor dem Vektor der auszugebenen Daten eingereiht. Somit erhält man auch unterschiedliche Datenstrukturen innerhalb der Objekte, was im folgenden Abschnitt näher betrachtet wird. Schließlich werden die in den *Results*-Objekten enthaltenen Daten analog zu den obigen Darstellungen in *Data*-Objekten gespeichert.

### C.3.3 Zugriff auf die Reportdaten

Um auf die in einem *Data*-Objekt enthaltenen Daten zuzugreifen, gibt es verschiedene Möglichkeiten. Hier wird oftmals die in den folgenden Codezeilen gezeigte Methode verwendet, wie sie bei Bodenkontakt- oder Sonnenlichtzeiten verwendet werden können.

```
start_t = Data.GetDataSetByName('Start Time').GetValues;  
stop_t  = Data.GetDataSetByName('Stop Time').GetValues;  
dur     = Data.GetDataSetByName('Duration').GetValues;
```

Darin werden die verschiedenen in *Data* gespeicherten Daten über ihre Bezeichnung aufgerufen. Dazu wird der Schnittstelle *GetDataSetByName* die Bezeichnung übergeben und die Methode *GetValues* darauf angewandt. Damit werden die entsprechenden Daten in Vektoren gespeichert, wobei zu beachten ist, dass es sich dabei zunächst um Vektoren handelt, deren Elemente sogenannte „Zellen“ sind. Eine solche Zelle (in MATLAB *Cell* genannt) kann eine beliebig große Matrix enthalten, weshalb nicht direkt mit den in diesem Fall skalaren Zahlenwerten weitergerechnet werden kann. Dazu kann entweder die MATLAB-Funktion *cell2mat* verwendet werden, mit der eine Matrix aus Zellen in eine „normale“ Matrix aus Zahlenwerten umgewandelt wird. Außerdem besteht die Möglichkeit, auf ein Element eines solchen Vektors zuzugreifen, indem der Index in geschweifte Klammern gesetzt wird. Damit wird direkt auf den in der Zelle



enthaltenen Zahlenwert zugegriffen [26].

Es existieren jedoch noch weitere Möglichkeiten, die in einem Objekt wie *Data* enthaltenen Daten zu erhalten, wie in der folgenden Zeile gezeigt.

```
vec = Data.ToArray;
```

Darin werden alle Daten aus *Data* in einer Matrix *vec* gespeichert, wobei auch diese Matrix aus Zellen besteht, die zur weiteren Verwendung der Daten umgewandelt werden müssen. Dabei werden die verschiedenen Datensätze (also zum Beispiel Startzeitpunkt, Endzeitpunkt und Dauer) in verschiedenen Spalten und die verschiedenen Zeitpunkte (bzw. die verschiedenen Sonnenlicht- oder Bodenkontaktzeiten) in verschiedenen Zeilen gespeichert. Diese Art des Datenzugriffs ist sinnvoll, wenn das Solar Panel Tool (siehe Abschnitt 7.1.2) verwendet wird. Da dabei mehrere verschiedene Solarpaneele berücksichtigt werden, gibt es einen Datensatz (wie zum Beispiel die beleuchtete Fläche) für jedes Paneel. Mit der *GetDataSetByName*-Schnittstelle kann man dabei nur auf die Datensätze des ersten Paneels zugreifen. Die *ToArray*-Methode hingegen gibt eine Matrix mit allen Datensätzen aus.

Diese Methode kann jedoch nicht eingesetzt werden, wenn die verschiedenen Datensätze nicht die gleiche Länge besitzen. Dies ist zum Beispiel bei Berechnungen der CNR zu jedem Zeitpunkt einer Bodenkontaktphase der Fall (siehe auch Abschnitt C.3.2). Da diese Phasen nicht gleich lang sind, besitzen sie eine unterschiedliche Anzahl an diskreten Werten. Spalten kurzer Bodenkontaktzeiten mit wenig Elementen werden dann mit *not a number*-Werten (NaN) aufgefüllt. Die *ToArray*-Methode erzeugt aus den Daten eine Matrix, die nur so viele Spalten besitzt, wie die kürzeste Bodenkontaktzeit gültige Werte. Damit gehen bei längeren Bodenkontaktzeiten entsprechend Werte verloren.

In diesem Fall ist jedoch auch der Zugriff über die Bezeichnung nicht möglich, da dabei wieder nur auf die erste Bodenkontaktphase zugegriffen werden kann. Daher wird hier auf einzelne Zeilen der Daten (also die Zahlenwerte der verschiedenen Datensätze zu einem Zeitpunkt) zugegriffen, wie in der folgenden Codezeile dargestellt.

```
vec = Data.GetRow(0);
```

Darin wird die erste Zeile der Datensätze über die Methode *GetRow* im Vektor *vec* gespeichert, wobei beachtet werden muss, dass die Indizierung der Zeilen bei null beginnt. Diese Methode kann grundsätzlich für alle hier vorkommenden Fälle verwendet werden, jedoch muss dabei die Rechenzeit berücksichtigt werden. Jeder einzelne Zugriff auf das *Data*-Objekt ist verhältnismäßig rechenaufwändig, weshalb grundsätzlich angestrebt werden sollte, mit einem Befehl auf möglichst viele Daten auf einmal zuzugreifen. Die *GetRow*-Methode kann bei den Linkbudgetberechnungen problemlos verwendet werden, da dabei eine Matrix mit vielen Spalten (Bodenkontaktphasen) und relativ wenig Zeilen (Zeitpunkte in Abständen von einer Minute während einer Bodenkontaktphase) entsteht. Bei den Bodenkontaktzeiten ist dies genau umgekehrt: Die Datenmatrix enthält nur drei Spalten (Startzeitpunkt, End-

zeitpunkt und Dauer), aber über eine Missionsdauer von einem Jahr eine Vielzahl an Zeilen (Anzahl der Bodenkontaktzeiten). Daher ist dabei der Zugriff über die *GetDataSetByName* unverzichtbar.

Die *GetRow*-Methode kann darüber hinaus auch verwendet werden, um mit Hilfe der MATLAB-Funktion *length* die Anzahl der vorhandenen Spalten zu bestimmen. Die Anzahl der Zeilen kann bestimmt werden, indem die Methode *RowCount* auf das *Data*-Objekt angewendet wird.

## C.4 Verwendung des HPOP

Die Verwendung des HPOP für dies hier durchgeführten Simulationen wird in Abschnitt 8.2.3 beschrieben. Die zur Einstellung der Parameter notwendigen Codezeilen sind im Folgenden wiedergegeben.

```
Satellite.SetPropagatorType('ePropagatorHPOP');
Satellite.Propagator.StartTime = scen_start;
Satellite.Propagator.StopTime = scen_stop;
Satellite.Propagator.Step = 60;
Satellite.Propagator.InitialState.Representation.AssignClassical ...
... ('eCoordinateSystemJ2000',a,0.0,97.9908,0.0,293.06,45.0);
Satellite.Propagator.ForceModel.SolarRadiationPressure.SRPMModel.Model.Cr ...
... = 1.21;
Satellite.Propagator.ForceModel.SolarRadiationPressure.SRPMModel.Model. ...
... AreaMassRatio = 0.0104;
Satellite.Propagator.ForceModel.SolarRadiationPressure.UseBoundaryMitigation ...
... = 1;
Satellite.Propagator.ForceModel.Drag.AreaMassRatio = 0.0104;
Satellite.Propagator.ForceModel.Drag.SetSolarFluxGeoMagType ...
... ('eSolarFluxGeoMagUseFile');
Satellite.Propagator.ForceModel.Drag.SolarFluxGeoMag.File = ...
... 'C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\ ...
... AGI\STK 9\DynamicEarthData\stkFluxGeoMag.fxm';
Satellite.Propagator.ForceModel.MoreOptions.RadiationPressure.IncludeAlbedo ...
... = 1;
Satellite.Propagator.ForceModel.MoreOptions.RadiationPressure.IncludeThermal ...
... = 1;
Satellite.Propagator.ForceModel.MoreOptions.RadiationPressure.Ck = 1.21;
Satellite.Propagator.ForceModel.MoreOptions.RadiationPressure.AreaMassRatio ...
... = 0.021;
Satellite.Propagator.Propagate();
```

# Anhang D

## Grundlegende Skripte für die Simulationen

In diesem Abschnitt werden einige für die Simulationen grundlegende Skripte wiedergegeben. Dies sind zunächst die \*.mdl-Dateien, mit denen in STK die Satellitenmodelle mit den Solarpaneelen für das Solar Panel Tool definiert werden (siehe Abschnitt 7.1.2). Darüber hinaus wird die MATLAB-Funktion zur Generierung der \*.a-Dateien zur Definition der Lage in STK wiedergegeben (siehe Abschnitt 8.2.2). Außerdem wird das MATLAB-Skript zum Vergleich der beiden Berechnungsmethoden der Leistung der Solarpaneele gezeigt (siehe Abschnitt 8.3.3). Dabei werden zu lange Zeilen wieder umgebrochen und entsprechend drei Punkte gesetzt (siehe Abschnitt C.2.4).

### D.1 \*.mdl-Dateien

#### Standardkonfiguration: CLAVIS\_Standard.mdl

```
# Solar Panel Definition #####
Component panel1 # upper panel
  SolarPanel CLAVIS_up
  Polygon
    FaceColor %002004040
    Shininess 128
    Specularity 0.9
    NumVerts 4
    Data
# sequence of Data-points is crucial! If order is reversed,
# panel is assumed to face inwards -> no power generation!
    0.13 0.13 -0.305 # front right
    0.13 -0.13 -0.305 # front left
    -0.13 -0.13 -0.305 # back left
    -0.13 0.13 -0.305 # back right
  EndPolygon
EndComponent

Component panel2 # +x panel
  SolarPanel CLAVIS_+x
  Polygon
    FaceColor %002004040
    Shininess 128
    Specularity 0.9
    NumVerts 4
    Data
    0.175 -0.13 -0.26 #top left
    0.175 0.13 -0.26 #top right
    0.175 0.13 0.0 #bottom right
    0.175 -0.13 0.0 #bottom left
  EndPolygon
EndComponent

Component panel3 # -x panel
  SolarPanel CLAVIS_-x
```

```

    Polygon
        FaceColor %002004040
        Shininess 128
        Specularity 0.9
        NumVerts 4
        Data
            -0.175 -0.13 0.0      #bottom left
            -0.175 0.13 0.0      #bottom right
            -0.175 0.13 -0.26    #top right
            -0.175 -0.13 -0.26   #top left
    EndPolygon
EndComponent

Component panel4 # +y panel
SolarPanel CLAVIS_+y
Polygon
    FaceColor %002004040
    Shininess 128
    Specularity 0.9
    NumVerts 4
    Data
        -0.13 0.175 0.0      #bottom back
        0.13 0.175 0.0      #bottom front
        0.13 0.175 -0.26    #top front
        -0.13 0.175 -0.26   #top back
    EndPolygon
EndComponent

Component panel5 # -y panel
SolarPanel CLAVIS_-y
Polygon
    FaceColor %002004040
    Shininess 128
    Specularity 0.9
    NumVerts 4
    Data
        0.13 -0.175 0.0      #bottom front
        -0.13 -0.175 0.0     #bottom back
        -0.13 -0.175 -0.26   #top back
        0.13 -0.175 -0.26    #top front
    EndPolygon
EndComponent

Component panel
Refer
Component panel1
EndRefer
Refer
Component panel2
EndRefer
Refer
Component panel3
EndRefer
Refer
Component panel4
EndRefer
Refer
Component panel5
EndRefer
EndComponent

# END of Solar Panel Definition #####

##### BODY DEFINITION #####

Component body_up
FaceColor gray30
FaceStyle Filled
Polygon
    NumVerts 4
    Data
        -0.174 0.174 -0.304 # back right
        -0.174 -0.174 -0.304 # back left
        0.174 -0.174 -0.304 # front left
        0.174 0.174 -0.304  # front right
    EndPolygon
EndComponent

Component body_+x
FaceColor gray30
FaceStyle Filled
Polygon
    NumVerts 4
    Data
        0.174 -0.174 -0.304 #top left
        0.174 0.174 -0.304 #top right
        0.174 0.174 0.0     #bottom right
        0.174 -0.174 0.0    #bottom left
    EndPolygon

```

```

EndComponent

Component body_-x
  FaceColor gray30
  FaceStyle Filled
  Polygon
    NumVerts 4
    Data
      -0.174 -0.174 -0.304 #top left
      -0.174 0.174 -0.304 #top right
      -0.174 0.174 0.0 #bottom right
      -0.174 -0.174 0.0 #bottom left
    EndPolygon
  EndComponent

Component body_+y
  FaceColor gray30
  FaceStyle Filled
  Polygon
    NumVerts 4
    Data
      -0.174 0.174 0.0 #bottom back
      -0.174 0.174 -0.304 #top back
      0.174 0.174 -0.304 #top front
      0.174 0.174 0.0 #bottom front
    EndPolygon
  EndComponent

Component body_-y
  FaceColor gray30
  FaceStyle Filled
  Polygon
    NumVerts 4
    Data
      -0.174 -0.174 0.0 #bottom back
      -0.174 -0.174 -0.304 #top back
      0.174 -0.174 -0.304 #top front
      0.174 -0.174 0.0 #bottom front
    EndPolygon
  EndComponent

Component body_down
  FaceColor gray30
  FaceStyle Filled
  Polygon
    NumVerts 4
    Data
      -0.174 0.174 0 # back right
      -0.174 -0.174 0 # back left
      0.174 -0.174 0 # front left
      0.174 0.174 0 # front right
    EndPolygon
  EndComponent

Component launch_ring
  FaceColor gray30
  FaceStyle Filled
  Cylinder
    NumSides 16
    Face1Radius 0.1
    Face1Normal 1.0 0.0 0.0 # x-axis direction
    Face2Radius 0.1
    Face2Normal 1.0 0.0 0.0
    Length 0.04
  EndCylinder
  Rotate 0 90 0
  Translate 0 0 0.04
EndComponent

Component Body
  Refer
  Component body_up
  EndRefer
  Refer
  Component body_+x
  EndRefer
  Refer
  Component body_-x
  EndRefer
  Refer
  Component body_+y
  EndRefer
  Refer
  Component body_-y
  EndRefer
  Refer
  Component body_down
  EndRefer
  Refer

```

```

        Component launch_ring
    EndRefer
EndComponent

##### MODEL DEFINITION #####
Component CLAVIS
    Root
    Refer
        Component panel
    EndRefer
    Refer
        Component Body
    EndRefer
EndComponent

##### End of MODEL DEFINITION #####

```

### AISat-Konfiguration: CLAVIS\_AISat\_full.mdl

```

# Solar Panel Definition #####
Component panel1 # upper panel
    SolarPanel CLAVIS_up
    Polygon
        FaceColor %002004040
        Shininess 128
        Specularity 0.9
        NumVerts 4
        Data
# sequence of Data-points is crucial! If order is reversed,
# panel is assumed to face inwards -> no power generation!
        0.13 0.13 -0.29 # front right
        0.13 -0.13 -0.29 # front left
        -0.13 -0.13 -0.29 # back left
        -0.13 0.13 -0.29 # back right
    EndPolygon
EndComponent

Component panel2 # +x panel
    SolarPanel CLAVIS_+x
    Polygon
        FaceColor %002004040
        Shininess 128
        Specularity 0.9
        NumVerts 4
        Data
        0.16 -0.13 -0.26 #top left
        0.16 0.13 -0.26 #top right
        0.16 0.13 0.0 #bottom right
        0.16 -0.13 0.0 #bottom left
    EndPolygon
EndComponent

Component panel3 # -x panel
    SolarPanel CLAVIS_-x
    Polygon
        FaceColor %002004040
        Shininess 128
        Specularity 0.9
        NumVerts 4
        Data
        -0.16 -0.13 0.0 #bottom left
        -0.16 0.13 0.0 #bottom right
        -0.16 0.13 -0.26 #top right
        -0.16 -0.13 -0.26 #top left
    EndPolygon
EndComponent

Component panel4 # +y panel
    SolarPanel CLAVIS_+y
    Polygon
        FaceColor %002004040
        Shininess 128
        Specularity 0.9
        NumVerts 4
        Data
        -0.13 0.16 0.0 #bottom back
        0.13 0.16 0.0 #bottom front
        0.13 0.16 -0.26 #top front
        -0.13 0.16 -0.26 #top back
    EndPolygon
EndComponent

Component panel5 # -y panel
    SolarPanel CLAVIS_-y
    Polygon

```

```

    FaceColor %002004040
    Shininess 128
    Specularity 0.9
    NumVerts 4
    Data
    0.13 -0.16 0.0      #bottom front
    -0.13 -0.16 0.0    #bottom back
    -0.13 -0.16 -0.26 #top back
    0.13 -0.16 -0.26  #top front

EndPolygon
EndComponent

Component panel
Refer
Component panel1
EndRefer
Refer
Component panel2
EndRefer
Refer
Component panel3
EndRefer
Refer
Component panel4
EndRefer
Refer
Component panel5
EndRefer
EndComponent
# END of Solar Panel Definition #####

##### MISSION PANEL DEFINITION #####
Component Mission
    FaceColor gray20
    FaceStyle Filled
    Cylinder
        NumSides 64
        FacelRadius 0.295
        FacelNormal 0.0 0.0 0.0 # x-axis direction
        Face2Radius 0.295
        Face2Normal 0.0 0.0 0.0
        Length 0.02
    EndCylinder
    Rotate 0 90 0 # y-axis rotation 90-deg
    Translate 0.0 0.0 0.02
EndComponent
##### End of MISSION PANEL DEFINITION #####

##### High Gain Antenna Definition
Component Antenna
    FaceColor gray30
    FaceStyle Filled
    Helix
        NumSides 16
        NumCoils 8
        CoilHeight 2.5
        CoilRadius 0.285
        WireRadius 1
    EndHelix
    Rotate 0 270 0
    Translate 0.0 0.0 0.02
EndComponent
##### end of high gain antenna definition

##### reflector definition

# Radial Rods
Component rod1
    FaceColor gray30
    FaceStyle Filled
    Extrusion
        Length 0.63
        NumVerts 4
        Data
        0 0 0
        0 0.001 0
        0 0.001 0.015
        0 0 0.015
    EdgeLines
    EndExtrusion
EndComponent

Component rod2
    FaceColor gray30
    FaceStyle Filled
    Extrusion
        Length 0.63
        NumVerts 4

```

```

    Data
    0 0 0
    0 0.001 0
    0 0.001 0.015
    0 0 0.015
    EdgeLines
    EndExtrusion
    Rotate 0 0 45
EndComponent

Component rod3
FaceColor gray30
FaceStyle Filled
Extrusion
Length 0.63
NumVerts 4
Data
0 0 0
0 0.001 0
0 0.001 0.015
0 0 0.015
EdgeLines
EndExtrusion
Rotate 0 0 90
EndComponent

Component rod4
FaceColor gray30
FaceStyle Filled
Extrusion
Length 0.63
NumVerts 4
Data
0 0 0
0 0.001 0
0 0.001 0.015
0 0 0.015
EdgeLines
EndExtrusion
Rotate 0 0 135
EndComponent

Component rod5
FaceColor gray30
FaceStyle Filled
Extrusion
Length 0.63
NumVerts 4
Data
0 0 0
0 0.001 0
0 0.001 0.015
0 0 0.015
EdgeLines
EndExtrusion
Rotate 0 0 180
EndComponent

Component rod6
FaceColor gray30
FaceStyle Filled
Extrusion
Length 0.63
NumVerts 4
Data
0 0 0
0 0.001 0
0 0.001 0.015
0 0 0.015
EdgeLines
EndExtrusion
Rotate 0 0 225
EndComponent

Component rod7
FaceColor gray30
FaceStyle Filled
Extrusion
Length 0.63
NumVerts 4
Data
0 0 0
0 0.001 0
0 0.001 0.015
0 0 0.015
EdgeLines
EndExtrusion
Rotate 0 0 270
EndComponent

```



```

Component rod8
  FaceColor gray30
  FaceStyle Filled
  Extrusion
    Length 0.63
    NumVerts 4
    Data
      0 0 0
      0 0.001 0
      0 0.001 0.015
      0 0 0.015
    EdgeLines
  EndExtrusion
  Rotate 0 0 315
EndComponent

# tangential large rods

Component tlrod1
  FaceColor gray30
  FaceStyle Filled
  Extrusion
    Length 0.4822
    NumVerts 4
    Data
      0 0 0
      0 0.001 0
      0 0.001 0.015
      0 0 0.015
    EdgeLines
  EndExtrusion
  Rotate 0 0 112.5
  Translate 0.63 0 0
EndComponent

Component tlrod2
  FaceColor gray30
  FaceStyle Filled
  Extrusion
    Length 0.4822
    NumVerts 4
    Data
      0 0 0
      0 0.001 0
      0 0.001 0.015
      0 0 0.015
    EdgeLines
  EndExtrusion
  Rotate 0 0 157.5
  Translate 0.445 0.445 0
EndComponent

Component tlrod3
  FaceColor gray30
  FaceStyle Filled
  Extrusion
    Length 0.4822
    NumVerts 4
    Data
      0 0 0
      0 0.001 0
      0 0.001 0.015
      0 0 0.015
    EdgeLines
  EndExtrusion
  Rotate 0 0 202.5
  Translate 0 0.63 0
EndComponent

Component tlrod4
  FaceColor gray30
  FaceStyle Filled
  Extrusion
    Length 0.4822
    NumVerts 4
    Data
      0 0 0
      0 0.001 0
      0 0.001 0.015
      0 0 0.015
    EdgeLines
  EndExtrusion
  Rotate 0 0 247.5
  Translate -0.445 0.445 0
EndComponent

Component tlrod5
  FaceColor gray30

```

```

FaceStyle Filled
Extrusion
  Length 0.4822
  NumVerts 4
  Data
    0 0 0
    0 0.001 0
    0 0.001 0.015
    0 0 0.015
  EdgeLines
EndExtrusion
Rotate 0 0 292.5
Translate -0.63 0 0
EndComponent

Component tlrod6
FaceColor gray30
FaceStyle Filled
Extrusion
  Length 0.4822
  NumVerts 4
  Data
    0 0 0
    0 0.001 0
    0 0.001 0.015
    0 0 0.015
  EdgeLines
EndExtrusion
Rotate 0 0 337.5
Translate -0.445 -0.445 0
EndComponent

Component tlrod7
FaceColor gray30
FaceStyle Filled
Extrusion
  Length 0.4822
  NumVerts 4
  Data
    0 0 0
    0 0.001 0
    0 0.001 0.015
    0 0 0.015
  EdgeLines
EndExtrusion
Rotate 0 0 22.5
Translate 0 -0.63 0
EndComponent

Component tlrod8
FaceColor gray30
FaceStyle Filled
Extrusion
  Length 0.4822
  NumVerts 4
  Data
    0 0 0
    0 0.001 0
    0 0.001 0.015
    0 0 0.015
  EdgeLines
EndExtrusion
Rotate 0 0 67.5
Translate 0.445 -0.445 0
EndComponent

# tangential small rods

Component tsrod1
FaceColor gray30
FaceStyle Filled
Extrusion
  Length 0.3597
  NumVerts 4
  Data
    0 0 0
    0 0.001 0
    0 0.001 0.015
    0 0 0.015
  EdgeLines
EndExtrusion
Rotate 0 0 112.5
Translate 0.47 0 0
EndComponent

Component tsrod2
FaceColor gray30
FaceStyle Filled
Extrusion

```

```

        Length 0.3597
        NumVerts 4
        Data
        0 0 0
        0 0.001 0
        0 0.001 0.015
        0 0 0.015
        EdgeLines
    EndExtrusion
    Rotate 0 0 157.5
    Translate 0.3323 0.3323 0
EndComponent

Component tsrod3
    FaceColor gray30
    FaceStyle Filled
    Extrusion
        Length 0.3597
        NumVerts 4
        Data
        0 0 0
        0 0.001 0
        0 0.001 0.015
        0 0 0.015
        EdgeLines
    EndExtrusion
    Rotate 0 0 202.5
    Translate 0 0.47 0
EndComponent

Component tsrod4
    FaceColor gray30
    FaceStyle Filled
    Extrusion
        Length 0.3597
        NumVerts 4
        Data
        0 0 0
        0 0.001 0
        0 0.001 0.015
        0 0 0.015
        EdgeLines
    EndExtrusion
    Rotate 0 0 247.5
    Translate -0.3323 0.3323 0
EndComponent

Component tsrod5
    FaceColor gray30
    FaceStyle Filled
    Extrusion
        Length 0.3597
        NumVerts 4
        Data
        0 0 0
        0 0.001 0
        0 0.001 0.015
        0 0 0.015
        EdgeLines
    EndExtrusion
    Rotate 0 0 292.5
    Translate -0.47 0 0
EndComponent

Component tsrod6
    FaceColor gray30
    FaceStyle Filled
    Extrusion
        Length 0.3597
        NumVerts 4
        Data
        0 0 0
        0 0.001 0
        0 0.001 0.015
        0 0 0.015
        EdgeLines
    EndExtrusion
    Rotate 0 0 337.5
    Translate -0.3323 -0.3323 0
EndComponent

Component tsrod7
    FaceColor gray30
    FaceStyle Filled
    Extrusion
        Length 0.3597
        NumVerts 4
        Data
        0 0 0

```

```

    0 0.001 0
    0 0.001 0.015
    0 0 0.015
    EdgeLines
EndExtrusion
Rotate 0 0 22.5
Translate 0 -0.47 0
EndComponent

Component tsrod8
FaceColor gray30
FaceStyle Filled
Extrusion
Length 0.3597
NumVerts 4
Data
0 0 0
0 0.001 0
0 0.001 0.015
0 0 0.015
EdgeLines
EndExtrusion
Rotate 0 0 67.5
Translate 0.3323 -0.3323 0
EndComponent

# complete component

Component Reflector
Refer
Component rod1
EndRefer
Refer
Component rod2
EndRefer
Refer
Component rod3
EndRefer
Refer
Component rod4
EndRefer
Refer
Component rod5
EndRefer
Refer
Component rod6
EndRefer
Refer
Component rod7
EndRefer
Refer
Component rod8
EndRefer
Refer
Component tsrod1
EndRefer
Refer
Component tsrod2
EndRefer
Refer
Component tsrod3
EndRefer
Refer
Component tsrod4
EndRefer
Refer
Component tsrod5
EndRefer
Refer
Component tsrod6
EndRefer
Refer
Component tsrod7
EndRefer
Refer
Component tsrod8
EndRefer
Refer
Component tlrod1
EndRefer
Refer
Component tlrod2
EndRefer
Refer
Component tlrod3
EndRefer
Refer
Component tlrod4
EndRefer

```

```

Refer
  Component tlrod5
EndRefer
Refer
  Component tlrod6
EndRefer
Refer
  Component tlrod7
EndRefer
Refer
  Component tlrod8
EndRefer
EndComponent

##### end of reflector definition

##### low gain antenna definition
Component ant1
  FaceColor gray30
  FaceStyle Filled
  Extrusion
    Length 0.17
    NumVerts 4
    Data
      0 -0.0115 0.003
      0 -0.0115 -0.003
      0 0.0115 -0.003
      0 0.0115 0.003
    EdgeLines
  EndExtrusion
  Rotate 0 45 45
  Translate 0.194 0.194 0
EndComponent

Component ant2
  FaceColor gray30
  FaceStyle Filled
  Extrusion
    Length 0.495
    NumVerts 4
    Data
      0 -0.0115 0.003
      0 -0.0115 -0.003
      0 0.0115 -0.003
      0 0.0115 0.003
    EdgeLines
  EndExtrusion
  Rotate 0 45 -45
  Translate 0.194 -0.194 0
EndComponent

Component ant3
  FaceColor gray30
  FaceStyle Filled
  Extrusion
    Length 0.495
    NumVerts 4
    Data
      0 -0.0115 0.003
      0 -0.0115 -0.003
      0 0.0115 -0.003
      0 0.0115 0.003
    EdgeLines
  EndExtrusion
  Rotate 0 45 135
  Translate -0.194 0.194 0
EndComponent

Component ant4
  FaceColor gray30
  FaceStyle Filled
  Extrusion
    Length 0.17
    NumVerts 4
    Data
      0 -0.0115 0.003
      0 -0.0115 -0.003
      0 0.0115 -0.003
      0 0.0115 0.003
    EdgeLines
  EndExtrusion
  Rotate 0 135 45
  Translate -0.194 -0.194 0
EndComponent

Component LGAntennas
  Refer
    Component ant1
  EndRefer

```

```

Refer
  Component ant2
EndRefer
Refer
  Component ant3
EndRefer
Refer
  Component ant4
EndRefer
EndComponent

##### end of low gain antenna definition

##### body definition

Component body_up
  FaceColor gray30
  FaceStyle Filled
  Polygon
    NumVerts 4
    Data
      -0.159 0.159 -0.289 # back right
      -0.159 -0.159 -0.289 # back left
      0.159 -0.159 -0.289 # front left
      0.159 0.159 -0.289 # front right
    EndPolygon
  EndComponent

Component body_+x
  FaceColor gray30
  FaceStyle Filled
  Polygon
    NumVerts 4
    Data
      0.159 -0.159 -0.289 #top left
      0.159 0.159 -0.289 #top right
      0.159 0.159 0.0 #bottom right
      0.159 -0.159 0.0 #bottom left
    EndPolygon
  EndComponent

Component body_-x
  FaceColor gray30
  FaceStyle Filled
  Polygon
    NumVerts 4
    Data
      -0.159 -0.159 -0.289 #top left
      -0.159 0.159 -0.289 #top right
      -0.159 0.159 0.0 #bottom right
      -0.159 -0.159 0.0 #bottom left
    EndPolygon
  EndComponent

Component body_+y
  FaceColor gray30
  FaceStyle Filled
  Polygon
    NumVerts 4
    Data
      -0.159 0.159 0.0 #bottom back
      -0.159 0.159 -0.289 #top back
      0.159 0.159 -0.289 #top front
      0.159 0.159 0.0 #bottom front
    EndPolygon
  EndComponent

Component body_-y
  FaceColor gray30
  FaceStyle Filled
  Polygon
    NumVerts 4
    Data
      -0.159 -0.159 0.0 #bottom back
      -0.159 -0.159 -0.289 #top back
      0.159 -0.159 -0.289 #top front
      0.159 -0.159 0.0 #bottom front
    EndPolygon
  EndComponent

Component Body
  Refer
    Component body_up
  EndRefer

  Refer
    Component body_+x
  EndRefer

```

```

Refer
Component body_-x
EndRefer

Refer
Component body_+y
EndRefer

Refer
Component body_-y
EndRefer
EndComponent

##### end of body definition

##### MODEL DEFINITION #####
Component CLAVIS
Root
Refer
Component panel
EndRefer
Refer
Component Mission
EndRefer
Refer
Component Antenna
EndRefer
Refer
Component Reflector
EndRefer
Refer
Component LGAntennas
EndRefer
Refer
Component Body
EndRefer
EndComponent
##### End of MODEL DEFINITION #####

```

## D.2 attitude\_file\_generator.m

Zweck: Erstellen einer \*.a Datei für gegebene Amplituden der Roll- und Nickschwingungen um die Nadirrichtung über die gesamte Missionsdauer

```

function [out]=attitude_file_generator(epoch,pitch_ampl,roll_ampl,rot_angle,file_path,a,dur)

% create an attitude-file (*.a) in the specified folder for STK using MATLAB
% with pitch and roll oscillations, frequency: 2 per orbit
% input:
% epoch as UTCG string
% pitch oscillation amplitude in deg
% roll oscillation amplitude in deg
% rotation angle 0°/45°
% file path and name as string
% semi major axis in metres
% duration of the mission in seconds

tU = 2*pi*sqrt(a^3/3.986e14);

%calculate YPR angles
time = 0:(0.01*tU):dur+86400;
l = length(time);
input_file = zeros(l,4);
for i = 1:l
    input_file(i,1) = time(i);
    input_file(i,2) = rot_angle; % yaw
    if input_file(i,2) > 360
        input_file(i,2) = input_file(i,2)-360;
    end
    input_file(i,3) = pitch_ampl*cos(time(i)*4*pi/tU); % pitch
    input_file(i,4) = roll_ampl*sin(time(i)*4*pi/tU); % roll
end

% open file
fid = fopen(file_path,'w');

% print stuff to file
fprintf(fid, 'stk.v.5.0 \r\n\r\n');
fprintf(fid, 'BEGIN Attitude \r\n\r\n');
fprintf(fid, 'ScenarioEpoch \t\t %s \r\n', epoch);
fprintf(fid, 'NumberOfAttitudePoints \t %u \r\n', l);
fprintf(fid, 'BlockingFactor \t\t 20 \r\n');
fprintf(fid, 'InterpolationOrder \t 1 \r\n');

```

```

fprintf(fid, 'CentralBody \t\t Earth \r\n');
fprintf(fid, 'CoordinateAxes \t\t Custom VVLH(Earth) Satellite/CLAVIS \r\n');
fprintf(fid, 'Sequence \t\t 321 \r\n\r\n');
fprintf(fid, 'AttitudeTimeYPRAngles \r\n\r\n');
fprintf(fid, '%f \t %f \t %f \t %f \r\n', input_file');
fprintf(fid, '\r\n END Attitude \r\n');

% close file
fclose(fid);

out = 1;
end

```

## D.3 Skript zum Vergleich der Berechnungsmethoden der elektrischen Leistung

### compare\_num\_ana\_energy\_STK.m

Zweck: Simulation der durch die verschiedenen Paneele der Standardkonfiguration bereitgestellten Leistung über einen Orbit, Berechnung analytisch und numerisch durch das Solar Panel Tool

```

clear all
close all
clc

% compare outputs of numerical calculation of power generation (STK Solar
% Panel Tool) and analytical calculation (panel normal vectors)

%-----
display('Hook to running STK...')
%-----
uiApp = actxGetRunningServer('STK9.application');
root = uiApp.Personality2;

% define units
root.UnitPreferences.SetCurrentUnit('DateFormat', 'UTC');
root.UnitPreferences.SetCurrentUnit('DistanceUnit', 'm');
root.UnitPreferences.SetCurrentUnit('TimeUnit', 'sec');

%-----
display('Create new objects and set their properties...')
%-----
% create new scenario, define scenario object
root.NewScenario('CLAVIS_Matlab_neu');
Scenario = root.GetObjectFromPath('*');
scen_start = '15 Aug 2011 00:00:00.000000000';
scen_stop = '15 Aug 2011 01:38:00.000000000';
Scenario.StopTime = scen_stop;
Scenario.StartTime = scen_start;

% remove standard 2D and 3D windows
root.ExecuteCommand('Window3D * Remove WindowID 1');
root.ExecuteCommand('Window2D * Remove WindowID 1');

% maximize and hide main window
root.ExecuteCommand('Application / Maximize');
root.ExecuteCommand('Application / Hide');

% create Satellite CLAVIS
Satellite = Scenario.Children.New('eSatellite', 'CLAVIS');
Satellite.SetPropagatorType('ePropagatorJ2Perturbation');
Satellite.Propagator.UseScenarioAnalysisTime;
Satellite.Propagator.InitialState.Representation.AssignClassical ...
... ('eCoordinateSystemJ2000', 7028140, 0.0, 97.9908, 0.0, 293.06, 0.0);
Satellite.Propagator.Propagate();

% set attitude and model file for satellite
Satellite.SetAttitudeType('eAttitudeStandard');
Satellite.Attitude.Basic.SetProfileType('eProfileSpinAboutNadir');
Satellite.Attitude.Basic.Profile.Rate = 0;
Satellite.Attitude.Basic.Profile.Offset = 0;
Satellite.Attitude.Basic.Profile.Epoch = '15 Aug 2011 00:00:00.0';

Satellite.VO.Model.ModelType = 'eModelFile';
Satellite.VO.Model.ModelData.Filename = 'STKData\VO\Models\Space\CLAVIS_Standard.mdl';

% only Earth as eclipse body
if Satellite.EclipseBodies.IsCentralBodyAssigned('Moon') == 1
    Satellite.EclipseBodies.UseCustomizedList = 1;
end

```



```

    Satellite.EclipseBodies.AssignCentralBody('Earth');
end

% time step in sec
time_step = 60;
time_step_string = int2str(time_step);

%-----
display('Get lighting time...')
%-----
% get lighting times
DataProvider = Satellite.DataProviders.Item('Lighting Times').Group.Item('Sunlight');
RptElems = {'Start Time'; 'Stop Time'; 'Duration'};
Results = DataProvider.ExecElements(Scenario.StartTime, Scenario.StopTime, RptElems);
Data = Results.DataSets;

% save data in vectors
num_light_per = Data.RowCount;
start_time = repmat('00 Jan 0000 00:00:00.000000000', num_light_per, 1);
stop_time = repmat('00 Jan 0000 00:00:00.000000000', num_light_per, 1);
start_epmin = zeros(num_light_per, 1);
stop_epmin = zeros(num_light_per, 1);
duration = zeros(num_light_per, 1);
start_t = Data.GetDataSetByName('Start Time').GetValues;
stop_t = Data.GetDataSetByName('Stop Time').GetValues;
dur = Data.GetDataSetByName('Duration').GetValues;
for i = 1:num_light_per
    if length(start_t{i}) == 30
        start_time(i,:) = start_t{i};
    else
        start_time(i,:) = ['0', start_t{i}];
    end
    start_epmin(i) = UTCG2EpMin(scen_start, start_time(i,:));
    if length(stop_t{i}) == 30
        stop_time(i,:) = stop_t{i};
    else
        stop_time(i,:) = ['0', stop_t{i}];
    end
    stop_epmin(i) = UTCG2EpMin(scen_start, stop_time(i,:));
    duration(i) = dur{i};
end

%-----
display('Compute power numerically for daylight period...')
%-----
% new date format
root.UnitPreferences.SetCurrentUnit('DateFormat', 'EpMin');
% Add all Solar Panel groups, Radius is crucial!!!
root.ExecuteCommand('VO */Satellite/CLAVIS SolarPanel Visualization AddGroup CLAVIS_up ...
... AddGroup CLAVIS_+x AddGroup CLAVIS_-x AddGroup CLAVIS_+y AddGroup CLAVIS_-y ...
... View On Radius On 0.5');
% set offscreen rendering on, view must be on in VO SolarPanel Visualisation
root.ExecuteCommand('Window3D * SetRenderMethod Method PBuffer WindowID 1');
root.ExecuteCommand('Window3D * Maximize WindowID 1');
% prepare report writing
DataProvider_power = Satellite.DataProviders.Item('Solar Panel Area');
ReportElements_power = {'Time'; 'Area'; 'Effective Area'; 'Solar Intensity'};
% create string for Solar Panel Exposure calculations
comm_str = ['VO */Satellite/CLAVIS SolarPanel Compute "' start_time "' " stop_time "' ' ...
... time_step_string ' Area "D:\area_report.txt" ''];
% actually compute data
root.ExecuteCommand(comm_str);
% save report output
Results_power = DataProvider_power.ExecElements(start_epmin, stop_epmin, time_step, ...
... ReportElements_power);
Data_power = Results_power.DataSets;
% delete Solar panel data
root.ExecuteCommand('VO */Satellite/CLAVIS SolarPanel DeleteData');
% calculate solar power per unit area as function of time
sol_power = solar_power(start_epmin, stop_epmin, 227);
% calculate energy
% save data in matrix vec
vec_cell = Data_power.ToArray;
vec = zeros(Data_power.RowCount, length(Data_power.GetRow(1)));
for j1 = 1:Data_power.RowCount
    for j2 = 1:length(Data_power.GetRow(1))
        vec(j1, j2) = vec_cell{j1, j2};
    end
end
% save data in vectors
% general
time=zeros(Data_power.RowCount,1);
sun_intens=zeros(Data_power.RowCount,1);
power_total=zeros(Data_power.RowCount,1);
power_side=zeros(Data_power.RowCount,1);

% top panel
area_up=zeros(Data_power.RowCount,1);
effect_area_up=zeros(Data_power.RowCount,1);

```

```

incid_angle_up=zeros(Data_power.RowCount,1);
angle_factor_up=zeros(Data_power.RowCount,1);
power_up=zeros(Data_power.RowCount,1);

% + x panel
area_pl_x=zeros(Data_power.RowCount,1);
effect_area_pl_x=zeros(Data_power.RowCount,1);
incid_angle_pl_x=zeros(Data_power.RowCount,1);
angle_factor_pl_x=zeros(Data_power.RowCount,1);
power_pl_x=zeros(Data_power.RowCount,1);

% - x panel
area_mi_x=zeros(Data_power.RowCount,1);
effect_area_mi_x=zeros(Data_power.RowCount,1);
incid_angle_mi_x=zeros(Data_power.RowCount,1);
angle_factor_mi_x=zeros(Data_power.RowCount,1);
power_mi_x=zeros(Data_power.RowCount,1);

% + y panel
area_pl_y=zeros(Data_power.RowCount,1);
effect_area_pl_y=zeros(Data_power.RowCount,1);
incid_angle_pl_y=zeros(Data_power.RowCount,1);
angle_factor_pl_y=zeros(Data_power.RowCount,1);
power_pl_y=zeros(Data_power.RowCount,1);

% - y panel
area_mi_y=zeros(Data_power.RowCount,1);
effect_area_mi_y=zeros(Data_power.RowCount,1);
incid_angle_mi_y=zeros(Data_power.RowCount,1);
angle_factor_mi_y=zeros(Data_power.RowCount,1);
power_mi_y=zeros(Data_power.RowCount,1);

%loop over every time step
for j=1:(Data_power.RowCount)
    % general data
    time(j)=vec(j,1);
    sun_intens(j)=vec(j,4);

    % top panel
    if vec(j,2)>0.0676
        area_up(j)=0.0676;
    else
        area_up(j)=vec(j,2);
    end
    effect_area_up(j)=vec(j,3);
    incid_angle_up(j)=acos(effect_area_up(j)/area_up(j));
    if isnan(incid_angle_up(j))==1
        incid_angle_up(j)=pi/2;
    end
    if incid_angle_up(j)>(88.8*pi/180)
        angle_factor_up(j)=0;
    elseif incid_angle_up(j)<(88.8*pi/180) && incid_angle_up(j)>(50*pi/180)
        angle_factor_up(j)=-0.369*(cos(incid_angle_up(j)))^3+ ...
            ... 0.637*(cos(incid_angle_up(j)))^2+0.750*cos(incid_angle_up(j))-0.015;
    else
        angle_factor_up(j)=cos(incid_angle_up(j));
    end
    power_up(j)=sol_power*area_up(j)*sun_intens(j)*angle_factor_up(j)*0.1977*0.5513;

    % plus x panel
    if vec(j,6)>0.0676
        area_pl_x(j)=0.0676;
    else
        area_pl_x(j)=vec(j,6);
    end
    effect_area_pl_x(j)=vec(j,7);
    incid_angle_pl_x(j)=acos(effect_area_pl_x(j)/area_pl_x(j));
    if isnan(incid_angle_pl_x(j))==1
        incid_angle_pl_x(j)=pi/2;
    end
    if incid_angle_pl_x(j)>(88.8*pi/180)
        angle_factor_pl_x(j)=0;
    elseif incid_angle_pl_x(j)<(88.8*pi/180) && incid_angle_pl_x(j)>(50*pi/180)
        angle_factor_pl_x(j)=-0.369*(cos(incid_angle_pl_x(j)))^3+ ...
            ... 0.637*(cos(incid_angle_pl_x(j)))^2+0.750*cos(incid_angle_pl_x(j))-0.015;
    else
        angle_factor_pl_x(j)=cos(incid_angle_pl_x(j));
    end
    power_pl_x(j)=sol_power*area_pl_x(j)*sun_intens(j)*angle_factor_pl_x(j)*0.1977*0.5513;

    % minus x panel
    if vec(j,10)>0.0676
        area_mi_x(j)=0.0676;
    else
        area_mi_x(j)=vec(j,10);
    end
    effect_area_mi_x(j)=vec(j,11);
    incid_angle_mi_x(j)=acos(effect_area_mi_x(j)/area_mi_x(j));
    if isnan(incid_angle_mi_x(j))==1

```

```

        incid_angle_mi_x(j)=pi/2;
    end
    if incid_angle_mi_x(j)>(88.8*pi/180)
        angle_factor_mi_x(j)=0;
    elseif incid_angle_mi_x(j)<(88.8*pi/180) && incid_angle_mi_x(j)>(50*pi/180)
        angle_factor_mi_x(j)=-0.369*(cos(incid_angle_mi_x(j)))^3+ ...
        ... 0.637*(cos(incid_angle_mi_x(j)))^2+0.750*cos(incid_angle_mi_x(j))-0.015;
    else
        angle_factor_mi_x(j)=cos(incid_angle_mi_x(j));
    end
    power_mi_x(j)=sol_power*area_mi_x(j)*sun_intens(j)*angle_factor_mi_x(j)*0.1977*0.5513;

    % plus y panel
    if vec(j,14)>0.0676
        area_pl_y(j)=0.0676;
    else
        area_pl_y(j)=vec(j,14);
    end
    effect_area_pl_y(j)=vec(j,15);
    incid_angle_pl_y(j)=acos(effect_area_pl_y(j)/area_pl_y(j));
    if isnan(incid_angle_pl_y(j))==1
        incid_angle_pl_y(j)=pi/2;
    end
    if incid_angle_pl_y(j)>(88.8*pi/180)
        angle_factor_pl_y(j)=0;
    elseif incid_angle_pl_y(j)<(88.8*pi/180) && incid_angle_pl_y(j)>(50*pi/180)
        angle_factor_pl_y(j)=-0.369*(cos(incid_angle_pl_y(j)))^3+ ...
        ... 0.637*(cos(incid_angle_pl_y(j)))^2+0.750*cos(incid_angle_pl_y(j))-0.015;
    else
        angle_factor_pl_y(j)=cos(incid_angle_pl_y(j));
    end
    power_pl_y(j)=sol_power*area_pl_y(j)*sun_intens(j)*angle_factor_pl_y(j)*0.1977*0.5513;

    % minus y panel
    if vec(j,18)>0.0676
        area_mi_y(j)=0.0676;
    else
        area_mi_y(j)=vec(j,18);
    end
    effect_area_mi_y(j)=vec(j,19);
    incid_angle_mi_y(j)=acos(effect_area_mi_y(j)/area_mi_y(j));
    if isnan(incid_angle_mi_y(j))==1
        incid_angle_mi_y(j)=pi/2;
    end
    if incid_angle_mi_y(j)>(88.8*pi/180)
        angle_factor_mi_y(j)=0;
    elseif incid_angle_mi_y(j)<(88.8*pi/180) && incid_angle_mi_y(j)>(50*pi/180)
        angle_factor_mi_y(j)=-0.369*(cos(incid_angle_mi_y(j)))^3+ ...
        ... 0.637*(cos(incid_angle_mi_y(j)))^2+0.750*cos(incid_angle_mi_y(j))-0.015;
    else
        angle_factor_mi_y(j)=cos(incid_angle_mi_y(j));
    end
    power_mi_y(j)=sol_power*area_mi_y(j)*sun_intens(j)*angle_factor_mi_y(j)*0.1977*0.5513;

    % total power
    power_total(j) = power_up(j)+power_pl_x(j)+power_mi_x(j)+power_pl_y(j)+power_mi_y(j);
    power_side(j) = power_pl_x(j)+power_mi_x(j)+power_pl_y(j)+power_mi_y(j);
end
power_total_num = power_total;
power_side_num = power_side;
power_up_num = power_up;
time_num = time;

%-----
display('Compute power analytically for daylight period...')
%-----
% vectors for panel normals in body frame, 0°, standard config
up_vec = [ 0 0 -1];
sq2h = sqrt(2)/2;
norm_vecs = zeros(4,3);
norm_vecs(1,:) = [ 1 0 0]; % +x
norm_vecs(2,:) = [-1 0 0]; % -x
norm_vecs(3,:) = [ 0 1 0]; % +y
norm_vecs(4,:) = [ 0 -1 0]; % -y
% Stefan's config 0°
norm_vecs_stef = zeros(4,3);
sq3h = sqrt(3)/2;
norm_vecs_stef(1,:) = [ 0.5 0 -sq3h]; % +x
norm_vecs_stef(2,:) = [-0.5 0 -sq3h]; % -x
norm_vecs_stef(3,:) = [ 0 0.5 -sq3h]; % +y
norm_vecs_stef(4,:) = [ 0 -0.5 -sq3h]; % -y

% solar panel area in m^2 (14 cells with 26.62cm^2 each)
pan_area = 0.037268;
% report content for power calc, unit sun vector in body frame
DataProvider_power = Satellite.DataProviders.Item('Vectors(Body)').Group.Item('Sun');
ReportElements_power = {'Time'; 'x/Magnitude'; 'y/Magnitude'; 'z/Magnitude'};

% save report output

```

```

Results_power = DataProvider_power.ExecElements(start_epmin, stop_epmin, time_step, ...
... ReportElements_power);
Data_power = Results_power.DataSets;

% calculate energy
% save data in matrix vec
time_vec = Data_power.GetDataSetByName('Time').GetValues;
xMag_vec = Data_power.GetDataSetByName('x/Magnitude').GetValues;
yMag_vec = Data_power.GetDataSetByName('y/Magnitude').GetValues;
zMag_vec = Data_power.GetDataSetByName('z/Magnitude').GetValues;
vec = zeros(Data_power.RowCount, length(Data_power.GetRow(1)));
for j = 1:Data_power.RowCount
    vec(j,1) = time_vec{j}(1);
    vec(j,2) = xMag_vec{j}(1);
    vec(j,3) = yMag_vec{j}(1);
    vec(j,4) = zMag_vec{j}(1);
end

% save data in vectors
% general
time = zeros(Data_power.RowCount,1);
power_total = zeros(Data_power.RowCount,1);
power_side = zeros(Data_power.RowCount,1);
power_total_stef = zeros(Data_power.RowCount,1);
power_side_stef = zeros(Data_power.RowCount,1);
sun_vector = zeros(Data_power.RowCount,3);

% top panel
incid_angle_up = zeros(Data_power.RowCount,1);
angle_factor_up = zeros(Data_power.RowCount,1);
power_up = zeros(Data_power.RowCount,1);
ang_sun_pan_up = zeros(Data_power.RowCount,1);

% side panels standard configuration
incid_angle = zeros(Data_power.RowCount,8);
angle_factor = zeros(Data_power.RowCount,8);
power = zeros(Data_power.RowCount,8);
ang_sun_pan = zeros(Data_power.RowCount,8);

% side panels Stefan's configuration
incid_angle_stef = zeros(Data_power.RowCount,8);
angle_factor_stef = zeros(Data_power.RowCount,8);
power_stef = zeros(Data_power.RowCount,8);
ang_sun_pan_stef = zeros(Data_power.RowCount,8);

for j=1:(Data_power.RowCount)
    % general data
    time(j) = vec(j,1);
    sun_vector(j,:) = vec(j,2:4);

    % top panel
    ang_sun_pan_up(j) = acos(dot(sun_vector(j,:),up_vec));
    if ang_sun_pan_up(j) > pi/2
        incid_angle_up(j) = pi/2;
    else
        incid_angle_up(j) = ang_sun_pan_up(j);
    end
    if incid_angle_up(j) > (88.8*pi/180)
        angle_factor_up(j) = 0;
    elseif incid_angle_up(j) < (88.8*pi/180) && incid_angle_up(j) > (50*pi/180)
        angle_factor_up(j) = -0.369*(cos(incid_angle_up(j)))^3+ ...
        ... 0.637*(cos(incid_angle_up(j)))^2+0.750*cos(incid_angle_up(j))-0.015;
    else
        angle_factor_up(j) = cos(incid_angle_up(j));
    end
    power_up(j) = sol_power*pan_area*angle_factor_up(j)*0.1977;

    % loop over side panels (4x0° spin offset, 4x45° spin offset)
    % standard configuration only for i=1 (minimum duration -> min
    % energy) and i=2 (average duration)
    for pan=1:4
        ang_sun_pan(j,pan) = acos(dot(sun_vector(j,:),norm_vecs(pan,:)));
        if ang_sun_pan(j,pan) > pi/2
            incid_angle(j,pan) = pi/2;
        else
            incid_angle(j,pan) = ang_sun_pan(j,pan);
        end
        if incid_angle(j,pan) > (88.8*pi/180)
            angle_factor(j,pan) = 0;
        elseif incid_angle(j,pan) < (88.8*pi/180) && incid_angle(j,pan) > (50*pi/180)
            angle_factor(j,pan) = -0.369*(cos(incid_angle(j,pan)))^3+ ...
            ... 0.637*(cos(incid_angle(j,pan)))^2+0.750*cos(incid_angle(j,pan))-0.015;
        else
            angle_factor(j,pan) = cos(incid_angle(j,pan));
        end
        power(j,pan) = sol_power*pan_area*angle_factor(j,pan)*0.1977;
    end

    % total power, side panels: average of 0° and 45° spin offset

```

```

power_total(j) = power_up(j)+sum(power(j,:));
power_side(j) = sum(power(j,:));

% loop over side panels (4x0° spin offset, 4x45° spin offset)
% Stefan's configuration only for i=2 (average case) and i=3
% (maximum duration smaller than tU -> min energy)
for pan=1:4
    ang_sun_pan_stef(j,pan) = acos(dot(sun_vector(j,:),norm_vecs_stef(pan,:)));
    if ang_sun_pan_stef(j,pan) > pi/2
        incid_angle_stef(j,pan) = pi/2;
    else
        incid_angle_stef(j,pan) = ang_sun_pan_stef(j,pan);
    end
    if incid_angle_stef(j,pan) > (88.8*pi/180)
        angle_factor_stef(j,pan) = 0;
    elseif incid_angle_stef(j,pan) < (88.8*pi/180) && incid_angle_stef(j,pan) > (50*pi/180)
        angle_factor_stef(j,pan) = -0.369*(cos(incid_angle_stef(j,pan)))^3+ ...
        ... 0.637*(cos(incid_angle_stef(j,pan)))^2+0.750*cos(incid_angle_stef(j,pan))-0.015;
    else
        angle_factor_stef(j,pan) = cos(incid_angle_stef(j,pan));
    end
    power_stef(j,pan) = sol_power*pan_area*angle_factor_stef(j,pan)*0.1977;
end

% total power, side panels: average of 0° and 45° spin offset
power_total_stef(j) = power_up(j)+sum(power_stef(j,:));
power_side_stef(j) = sum(power_stef(j,:));

end
power_total_ana = power_total;
power_side_ana = power_side;
power_up_ana = power_up;
time_ana = time;

power_total_ana_stef = power_total_stef;
power_side_ana_stef = power_side_stef;

plot(time_ana,power_total_ana,'k',time_ana,power_side_ana,'b',time_ana,power_up_ana,'r', ...
    ... time_num,power_total_num,'sk',time_num,power_side_num,'sk',time_num,power_up_num, ...
    ... 'sk','LineWidth',2,'MarkerSize',2)
set(gca,'FontName','Palatino','FontSize',12)
legend('gesamt analytisch','seitliche Paneele analytisch','oberes Paneel analytisch', ...
    ... 'numerische Simulation','Location','EastOutside')
xlabel('Zeit [min]')
xlim([11 76])
ylabel('Leistung [W]')
grid on

Scenario.Unload;

```

# Anhang E

## Simulation der AISat-Mission

In diesem Abschnitt wird zunächst das MATLAB-Skript zur Simulation der gesamten AISat-Mission detailliert beschrieben. Daraufhin werden die Skripte wiedergegeben, die zur Simulation der AISat-Mission und zur Auswertung der erhaltenen Daten verwendet werden (siehe Abschnitte 9.2 und 9.3). Abschließend werden die Skripte zur Simulation und zur Auswertung der Verläufe über einen Orbit wiedergegeben (siehe Abschnitt 9.1).

### E.1 Skript zur Simulation der AISat-Mission

Abbildung E.1 zeigt ein vereinfachtes Struktogramm (Nassi-Shneiderman-Diagramm) des MATLAB-Skripts, mit dem die Rohdaten für die AISat-Mission bestimmt wurden. Darin steht jede Zeile für mehrere Zeilen Code; hier soll vor allem die grobe Struktur des Skripts ersichtlich werden.

In dem Skript wird zunächst die Schnittstelle zwischen STK und MATLAB hergestellt, woraufhin die einzelnen Objekte definiert werden. Die dazu notwendigen Schritte wurden in Kapitel 7 und im Anhang C bereits ausführlich behandelt. Daraufhin werden der Anfangszeitpunkt, der Endzeitpunkt sowie die Dauer der einzelnen Sonnenlichtperioden bestimmt. Auch das dazu nötige Vorgehen wurde im Anhang C beschrieben. Dabei wird hier die Datumsangabe stets als Zeichenkette im UTCG-Format belassen, da dieses Format bei der Ausführung des Solar Panel Tools benötigt wird. Hierbei muss beachtet werden, dass die Länge der Zeichenkette bei Datumsangaben mit einstelligen Tagen ein Zeichen kürzer ist, als bei Datumsangaben mit zweistelligen Tagen. Um dadurch auftretende Probleme von vornherein auszuschließen, wird den kürzeren Datumsangaben eine Null vorangestellt.

Daraufhin werden Startzeitpunkt, Endzeitpunkt und Dauer der Bodenkontaktzeiten sowie das CNR in Zeitschritten von 60 Sekunden bestimmt und abgespeichert. Unter Berücksichtigung des in Abschnitt 5.3.3 definierten Mindestwertes für das CNR von 10 dB kann damit die Zeit bestimmt werden, während der es größer als dieser Mindestwert ist, um die effektiv nutzbare Bodenkontaktzeit zu erhalten. Das CNR ist dabei in erster Linie eine Funktion der Schrägentfernung und damit der Elevation und steigt mit der Elevation an. Daher gibt es zu Beginn und zum Ende jeder Bo-

Herstellen der Verbindung mit STK
Definieren der zu verwendenden Einheiten
Erstellen und Definieren des Szenarioobjekts
Manipulation der Fenster des STK-Anwendung
Erstellen und Definieren des Satellitenobjekts inklusive des Senderobjekts
Erstellen und Definieren des Bodenstationsobjekts inklusive des Empfängerobjekts
Definieren des verwendeten Zeitinkrements
Berechnen der Sonnenlichtzeiten (Startzeit, Endzeit, Dauer)
Abspeichern der Daten zu den Sonnenlichtzeiten
Berechnen der Bodenkontaktzeiten (Startzeit, Endzeit, Dauer) und des Linkbudgets (CNR)
Abspeichern der Bodenkontaktzeiten
Berechnen und Abspeichern der Zeitintervalle mit ausreichendem CNR
Initialisieren des Solar Panel Tools
Initialisieren der Variablen zur Speicherung der erzeugten Energie
Schleife über alle Sonnenlichtperioden
Berechnen der angestrahlten Fläche, der effektiv angestrahlten Fläche und der solaren Intensität zu jedem Zeitpunkt während der Sonnenlichtperiode für jedes Paneel mit Hilfe des Solar Panel Tools
Abspeichern der Daten in einer Matrix
Initialisieren der Variablen zur Berechnung der Leistung für jedes Paneel zu jedem Zeitpunkt
Schleife über alle Zeitpunkte während der Sonnenlichtperiode
Abspeichern der Zeit und der solaren Intensität zu dem Zeitpunkt
Berechnen der Leistung für die einzelnen Paneele zu dem Zeitpunkt
Berechnen der Gesamtleistung und Aufsummierung der Energie für den Zeitpunkt
Abspeichern der gesamten erzeugten Energie für die Sonnenlichtperiode
Ausgeben des Berechnungsfortschritts
Schließen des Fensters des Solar Panel Tools
Abspeichern der relevanten Daten als *.mat-Dateien auf der Festplatte
Schließen des Szenarios in STK

**Abbildung E.1:** Vereinfachtes Struktogramm eines MATLAB-Skripts zur Berechnung der Rohdaten zur Analyse der AISat-Mission

denkontaktphase einen Zeitraum, in dem das CNR unterhalb des Grenzwertes liegt. Wird ein bestimmter Abstand unterschritten und damit eine bestimmte Elevation überschritten, ist das CNR größer als 10 dB. Diese Dauer wird bestimmt, indem die Zeiträume zwischen zwei CNR-Werten größer als 10 dB aufsummiert werden. An den Übergängen wird die Zeit zwischen dem Wert größer als 10 dB und dem Wert kleiner als 10 dB durch lineare Interpolation bestimmt.

Im Skript wird daraufhin das Solar Panel Tool initialisiert (siehe Abschnitt 7.1.2). Dazu wird der Connect-Befehl *VO Solar Panel* verwendet. Damit werden zunächst die einzelnen Solarpaneele, die in der Modelldatei der AISat-Konfiguration einzeln definiert sind, hinzugefügt. Außerdem wird der Bound Radius auf 0,5m gesetzt. Daraufhin wird das Offscreen Rendering angeschaltet und das Fenster des Solar Panel Tools maximiert. Dies wird durch die folgenden Codezeilen realisiert, wobei die erste aufgrund ihrer Länge wieder umgebrochen wird (siehe Abschnitt C.2.4).

```
root.ExecuteCommand('VO */Satellite/CLAVIS SolarPanel Visualization ...
... AddGroup CLAVIS_up AddGroup CLAVIS_+x AddGroup CLAVIS_-x ...
... AddGroup CLAVIS_+y AddGroup CLAVIS_-y View On Radius On 0.5');
root.ExecuteCommand('Window3D * SetRenderMethod Method PBuffer WindowID 1');
root.ExecuteCommand('Window3D * Maximize WindowID 1');
```

Es folgt die Definition der auszugebenen Parameter. Diese sind hier die angestrahlte

Fläche für jedes Paneel, die effektiv angestrahlte Fläche für jedes Paneel, die Zeit sowie die „Solar Intensity“, die dem sichtbaren Anteil der Sonnenscheibe  $b_{\odot}$  entspricht. Daraufhin beginnt eine *for*-Schleife über alle Sonnenlichtperioden, wobei die erste und die letzte weggelassen werden, da diese durch die zeitliche Begrenzung der Mission unterbrochen sein können. In der Schleife soll für die aktuelle Sonnenlichtphase das Solar Panel Tool gestartet werden. Um Rechenzeit zu sparen und Überlappungen zu vermeiden, sollen dabei nur die Zeiträume simuliert werden, während der sich der Satellit tatsächlich in der Sonne befindet. Start- und Endzeitpunkte dieser Zeiträume wurden zu Beginn im UTCG-Format abgespeichert. Um nun Start- und Endzeitpunkt des Solar Panel Tools einstellen zu können, muss eine Zeichenkette erstellt werden, die als Eingabe dient. Diese beginnt wieder mit dem Aufruf des *VO Solar Panel*-Befehls mit dem Schlüsselwort *Compute* und enthält Start- und Endzeitpunkt im UTCG-Format, die in den Variablen *start\_time* und *stop\_time* gespeichert sind. Außerdem wird der Zeitschritt in Sekunden als Zeichenkette übergeben (*time\_step\_string*). Darüber hinaus muss eine Ausgabedatei definiert werden, die hier jedoch nicht weiter verwendet wird. Die erstellte Zeichenkette dient dann als Argument für den *root.ExecuteCommand*-Befehl, wie in den folgenden Codezeilen wiedergegeben. Damit wird das Solar Panel Tool gestartet und die vorher definierten Größen werden berechnet.

```
comm_str = ['VO */Satellite/CLAVIS SolarPanel Compute ...
... " ' start_time(i,:) ' " ' stop_time(i,:) ' " ' ...
... time_step_string ' Area "D:\area_report.txt" '];
root.ExecuteCommand(comm_str);
```

Die damit erhaltenen Daten werden daraufhin in eine Matrix eingelesen, wie auch in Abschnitt C.3.3 beschrieben wird. Außerdem werden Vektoren initialisiert, in denen die angestrahlte Fläche, die effektiv angestrahlte Fläche, der Einstrahlwinkel, der Verlustfaktor  $F_{incid}$  aus Gleichung (4.3) sowie die erzeugte Leistung zu jedem Zeitpunkt während der Sonnenlichtphase für jedes Paneel gespeichert werden. Anschließend beginnt eine weitere *for*-Schleife über alle Zeitpunkte der aktuell betrachteten Sonnenlichtphase. Innerhalb dieser Schleife wird für den aktuellen Zeitpunkt zunächst die Zeit und die Solar Intensity, die für die verschiedenen Paneele natürlich identisch sind, aus der Datenmatrix in dafür vorgesehene Vektoren eingelesen.

Daraufhin wird für jedes Paneel einzeln die erzeugte Leistung wie in Abschnitt 8.3.1 beschrieben zu dem aktuellen Zeitpunkt berechnet. Anschließend werden die von den einzelnen Paneelen erzeugten Leistungen zur Gesamtleistung zusammengerechnet. Daraufhin wird der Beitrag der zu diesem Zeitpunkt erzeugten Leistung zur Gesamtenergie über die Sonnenlichtphase bestimmt und die Energie aufsummiert. Damit endet die Schleife über die einzelnen Zeitpunkte einer Sonnenlichtphase. In der übergeordneten Schleife wird daraufhin die in dieser Sonnenlichtphase erzeugte Gesamtenergie für die seitlichen Paneele, das obere Paneel und für alle Paneele zusammen in den entsprechenden Vektoren gespeichert. Nach einer Ausgabe des aktuellen Rechenfortschritts endet auch diese Schleife.

Daraufhin wird das Fenster des Solar Panel Tools geschlossen und die entscheidenden Größen werden in \*.mat-Dateien mit Hilfe des *save*-Befehls gespeichert. Diese Dateien enthalten Startzeitpunkt, Endzeitpunkt und Dauer der Sonnenlichtphasen und der



Bodenkontaktzeiten sowie die Zeitspannen, während den das CNR größer als 10 dB ist. Außerdem werden die von den seitlichen Paneelen, dem oberen Paneel und allen Paneelen zusammen erzeugten Energien für jede Sonnenlichtphase gespeichert. Diese Dateien können in den Skripten zur Auswertung der Daten wieder geladen werden. Die entsprechenden Ausgabedateien finden sich auf der CD zur Arbeit.

Diese Trennung von Simulation und Auswertung hat den Vorteil, dass einmal generierte Rohdaten beliebig weiterverarbeitet werden können, ohne dass die aufwändige Simulation jedes Mal durchgeführt werden muss. Abschließend wird das Szenario in STK geschlossen. Das Skript ist als Funktion geschrieben, damit es aus einem Hauptprogramm mit anderen Skripten automatisch gestartet werden kann. Daher wird am Ende ein Ausgabeparameter auf eins gesetzt.

Grundsätzlich gibt es für jedes betrachtete Szenario der AISat-Mission zwei Skripte: Eines, in dem der Drehwinkel gleich  $0^\circ$  ist und eines in dem der Drehwinkel gleich  $45^\circ$  ist (siehe Abschnitt 8.2.1). Die in diesen Skripten erzeugten Daten können mit dem *load*-Befehl in MATLAB zur weiteren Analyse wieder geladen werden. In dem entsprechenden Auswertungsskript wird zunächst der Mittelwert der bei den beiden verschiedenen Drehwinkeln berechneten Energiewerte bestimmt. Die weitere Auswertung beruht auf den in den Abschnitten 4.4 und 5.3 gegebenen Gleichungen und ist ansonsten trivial. Das entsprechende Skript wird daher hier nicht weiter behandelt; es ist ebenfalls in den folgenden Abschnitten wiedergegeben.

### E.1.1 Quellcode

Das hier wiedergegebene Skript zur Simulation der AISat-Mission gilt für den Fall der Roll- und Nickschwingungen kleiner Amplitude bei einem Rotationswinkel von  $0^\circ$  unter Verwendung des  $J_2$ -Propagators. Die Skripte für die anderen Fälle sind dem hier gezeigten sehr ähnlich und unterscheiden sich lediglich in der Definition der Lage des Satelliten bzw. des verwendeten Propagators. Sie befinden sich auf der CD zur Arbeit.

#### CLAVIS\_AISat\_mission\_wobble0\_ampl5,3.m

```
function [out]=CLAVIS_AISat_mission_wobble0_ampl5,3()

%-----
display('Hook to running STK...')
%-----
uiApp    = actxGetRunningServer('STK9.application');
root     = uiApp.Personality2;

% define units
root.UnitPreferences.SetCurrentUnit('DateFormat', 'UTCG');
root.UnitPreferences.SetCurrentUnit('DistanceUnit', 'm');
root.UnitPreferences.SetCurrentUnit('TimeUnit', 'sec');
root.UnitPreferences.SetCurrentUnit('AngleUnit', 'deg');

%-----
display('Create new objects and set their properties...')
%-----
% create new scenario, define scenario object
root.NewScenario('CLAVIS_Matlab_neu');
Scenario = root.GetObjectFromPath('*/*');
scen_start = '15 Aug 2011 00:00:00.000000000';
scen_stop  = '15 Aug 2012 00:00:00.000000000';
Scenario.StopTime = scen_stop;
Scenario.StartTime = scen_start;

% remove standard 2D and 3D windows
root.ExecuteCommand('Window3D * Remove WindowID 1');
```

```

root.ExecuteCommand('Window2D * Remove WindowID 1');

% maximize and hide main window
root.ExecuteCommand('Application / Maximize');
root.ExecuteCommand('Application / Hide');

% create Satellite CLAVIS
Satellite = Scenario.Children.New('eSatellite', 'CLAVIS');
Satellite.SetPropagatorType('ePropagatorJ2Perturbation');
Satellite.Propagator.UseScenarioAnalysisTime;
Satellite.Propagator.InitialState.Representation.AssignClassical ...
... ('eCoordinateSystemJ2000', 7028137, 0.0, 97.9908, 0.0, 293.06, 0.0);
Satellite.Propagator.Propagate();

% create *.a file and set attitude file for satellite
file_path = 'G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\ ...
... wobble0_amp15,3\CLAVIS_wob5,3_0.a';
mission_dur = 366*24*3600;
attitude_file_generator(scen_start, 5, 3, 0, file_path, 7028140, mission_dur);
Satellite.SetAttitudeType('eAttitudeStandard');
Satellite.Attitude.External.Load(file_path);

% set model file
Satellite.VO.Model.ModelType = 'eModelFile';
Satellite.VO.Model.ModelData.FileName = 'STKData\VO\Models\Space\CLAVIS_AISat_full.mdl';

% only Earth as eclipse body
if Satellite.EclipseBodies.IsCentralBodyAssigned('Moon') == 1
    Satellite.EclipseBodies.UseCustomizedList = 1;
    Satellite.EclipseBodies.AssignCentralBody('Earth');
end

% time step in sec
time_step = 60;
time_step_string = int2str(time_step);

%-----
display('Get lighting times...')
%-----
% get lighting times
DataProvider = Satellite.DataProviders.Item('Lighting Times').Group.Item('Sunlight');
RptElems = {'Start Time'; 'Stop Time'; 'Duration'};
Results = DataProvider.ExecElements(Scenario.StartTime, Scenario.StopTime, RptElems);
Data = Results.DataSets;

% save data in vectors
num_light_per = Data.RowCount-2; % leave out 1st and last value
start_time = repmat('00 Jan 0000 00:00:00.000000000', num_light_per, 1);
stop_time = repmat('00 Jan 0000 00:00:00.000000000', num_light_per, 1);
start_epmin = zeros(num_light_per, 1);
stop_epmin = zeros(num_light_per, 1);
duration = zeros(num_light_per, 1);
start_t = Data.GetDataSetByName('Start Time').GetValues;
stop_t = Data.GetDataSetByName('Stop Time').GetValues;
dur = Data.GetDataSetByName('Duration').GetValues;
for i = 1:num_light_per
    if length(start_t{i+1}) == 30
        start_time(i,:) = start_t{i+1};
    else
        start_time(i,:) = ['0', start_t{i+1}];
    end
    start_epmin(i) = UTCG2EpMin(scen_start, start_time(i,:));
    if length(stop_t{i+1}) == 30
        stop_time(i,:) = stop_t{i+1};
    else
        stop_time(i,:) = ['0', stop_t{i+1}];
    end
    stop_epmin(i) = UTCG2EpMin(scen_start, stop_time(i,:));
    duration(i) = dur{i+1};
end

%-----
display('Compute power for each daylight period...')
%-----
% new date format
root.UnitPreferences.SetCurrentUnit('DateFormat', 'EpMin');
% Add all Solar Panel groups, Radius is crucial!!!
root.ExecuteCommand('VO */Satellite/CLAVIS SolarPanel Visualization AddGroup CLAVIS_up ...
... AddGroup CLAVIS_+x AddGroup CLAVIS_-x AddGroup CLAVIS_+y AddGroup CLAVIS_-y ...
... View On Radius On 0.5');
% set offscreen rendering on, view must be on in VO SolarPanel Visualisation
root.ExecuteCommand('Window3D * SetRenderMethod Method PBuffer WindowID 1');
root.ExecuteCommand('Window3D * Maximize WindowID 1');
% prepare report writing
DataProvider_power = Satellite.DataProviders.Item('Solar Panel Area');
ReportElements_power = {'Time'; 'Area'; 'Effective Area'; 'Solar Intensity'};
% preallocate vector for energy output
energy = zeros(length(duration), 1);
energy_top = zeros(length(duration), 1);

```

```

energy_side = zeros(length(duration),1);
% loop over all sunlight periods
for i = 1:length(duration)
    % create string for Solar Panel Exposure calculations
    comm_str = ['VO */Satellite/CLAVIS SolarPanel Compute "' start_time(i,:) '" "' ...
        ... stop_time(i,:) '" ' time_step_string ' Area "D:\area_report.txt" ''];
    % actually compute data
    root.ExecuteCommand(comm_str);
    % save report output
    Results_power = DataProvider_power.ExecElements(start_epmin(i), stop_epmin(i), ...
        ... time_step, ReportElements_power);
    Data_power = Results_power.DataSets;
    % delete Solar panel data
    root.ExecuteCommand('VO */Satellite/CLAVIS SolarPanel DeleteData');
    % calculate solar power per unit area as function of time
    sol_power = solar_power(start_epmin(i),stop_epmin(i),227);
    % calculate energy
    % save data in matrix vec
    vec_cell = Data_power.ToArray;
    vec = zeros(Data_power.RowCount,length(Data_power.GetRow(1)));
    for j1 = 1:Data_power.RowCount
        for j2 = 1:length(Data_power.GetRow(1))
            vec(j1,j2) = vec_cell{j1,j2};
        end
    end
    % save data in vectors
    % general
    time=zeros(Data_power.RowCount-1,1);
    sun_intens=zeros(Data_power.RowCount-1,1);
    power_total=zeros(Data_power.RowCount-1,1);
    power_side=zeros(Data_power.RowCount-1,1);

    % top panel
    area_up=zeros(Data_power.RowCount-1,1);
    effect_area_up=zeros(Data_power.RowCount-1,1);
    incid_angle_up=zeros(Data_power.RowCount-1,1);
    angle_factor_up=zeros(Data_power.RowCount-1,1);
    power_up=zeros(Data_power.RowCount-1,1);

    % + x panel
    area_pl_x=zeros(Data_power.RowCount-1,1);
    effect_area_pl_x=zeros(Data_power.RowCount-1,1);
    incid_angle_pl_x=zeros(Data_power.RowCount-1,1);
    angle_factor_pl_x=zeros(Data_power.RowCount-1,1);
    power_pl_x=zeros(Data_power.RowCount-1,1);

    % - x panel
    area_mi_x=zeros(Data_power.RowCount-1,1);
    effect_area_mi_x=zeros(Data_power.RowCount-1,1);
    incid_angle_mi_x=zeros(Data_power.RowCount-1,1);
    angle_factor_mi_x=zeros(Data_power.RowCount-1,1);
    power_mi_x=zeros(Data_power.RowCount-1,1);

    % + y panel
    area_pl_y=zeros(Data_power.RowCount-1,1);
    effect_area_pl_y=zeros(Data_power.RowCount-1,1);
    incid_angle_pl_y=zeros(Data_power.RowCount-1,1);
    angle_factor_pl_y=zeros(Data_power.RowCount-1,1);
    power_pl_y=zeros(Data_power.RowCount-1,1);

    % - y panel
    area_mi_y=zeros(Data_power.RowCount-1,1);
    effect_area_mi_y=zeros(Data_power.RowCount-1,1);
    incid_angle_mi_y=zeros(Data_power.RowCount-1,1);
    angle_factor_mi_y=zeros(Data_power.RowCount-1,1);
    power_mi_y=zeros(Data_power.RowCount-1,1);

    %loop over every time step
    en=0;
    en_up=0;
    en_side=0;
    for j=1:(Data_power.RowCount-1)
        % general data
        time(j)=vec(j,1);
        sun_intens(j)=vec(j,4);

        % top panel
        if vec(j,2)>0.0676
            area_up(j)=0.0676;
        else
            area_up(j)=vec(j,2);
        end
        effect_area_up(j)=vec(j,3);
        incid_angle_up(j)=acos(effect_area_up(j)/area_up(j));
        if isnan(incid_angle_up(j))==1
            incid_angle_up(j)=pi/2;
        end
        if incid_angle_up(j)>(88.8*pi/180)
            angle_factor_up(j)=0;
        end
    end
end

```

```

elseif incid_angle_up(j)<(88.8*pi/180) && incid_angle_up(j)>(50*pi/180)
    angle_factor_up(j)=-0.369*(cos(incid_angle_up(j)))^3+ ...
    ... 0.637*(cos(incid_angle_up(j)))^2+0.750*cos(incid_angle_up(j))-0.015;
else
    angle_factor_up(j)=cos(incid_angle_up(j));
end
power_up(j)=sol_power*area_up(j)*sun_intens(j)*angle_factor_up(j)*0.1977*0.5513;

% plus x panel
if vec(j,6)>0.0676
    area_pl_x(j)=0.0676;
else
    area_pl_x(j)=vec(j,6);
end
effect_area_pl_x(j)=vec(j,7);
incid_angle_pl_x(j)=acos(effect_area_pl_x(j)/area_pl_x(j));
if isnan(incid_angle_pl_x(j))==1
    incid_angle_pl_x(j)=pi/2;
end
if incid_angle_pl_x(j)>(88.8*pi/180)
    angle_factor_pl_x(j)=0;
elseif incid_angle_pl_x(j)<(88.8*pi/180) && incid_angle_pl_x(j)>(50*pi/180)
    angle_factor_pl_x(j)=-0.369*(cos(incid_angle_pl_x(j)))^3+ ...
    ... 0.637*(cos(incid_angle_pl_x(j)))^2+0.750*cos(incid_angle_pl_x(j))-0.015;
else
    angle_factor_pl_x(j)=cos(incid_angle_pl_x(j));
end
power_pl_x(j)=sol_power*area_pl_x(j)*sun_intens(j)*angle_factor_pl_x(j)*0.1977*0.5513;

% minus x panel
if vec(j,10)>0.0676
    area_mi_x(j)=0.0676;
else
    area_mi_x(j)=vec(j,10);
end
effect_area_mi_x(j)=vec(j,11);
incid_angle_mi_x(j)=acos(effect_area_mi_x(j)/area_mi_x(j));
if isnan(incid_angle_mi_x(j))==1
    incid_angle_mi_x(j)=pi/2;
end
if incid_angle_mi_x(j)>(88.8*pi/180)
    angle_factor_mi_x(j)=0;
elseif incid_angle_mi_x(j)<(88.8*pi/180) && incid_angle_mi_x(j)>(50*pi/180)
    angle_factor_mi_x(j)=-0.369*(cos(incid_angle_mi_x(j)))^3+ ...
    ... 0.637*(cos(incid_angle_mi_x(j)))^2+0.750*cos(incid_angle_mi_x(j))-0.015;
else
    angle_factor_mi_x(j)=cos(incid_angle_mi_x(j));
end
power_mi_x(j)=sol_power*area_mi_x(j)*sun_intens(j)*angle_factor_mi_x(j)*0.1977*0.5513;

% plus y panel
if vec(j,14)>0.0676
    area_pl_y(j)=0.0676;
else
    area_pl_y(j)=vec(j,14);
end
effect_area_pl_y(j)=vec(j,15);
incid_angle_pl_y(j)=acos(effect_area_pl_y(j)/area_pl_y(j));
if isnan(incid_angle_pl_y(j))==1
    incid_angle_pl_y(j)=pi/2;
end
if incid_angle_pl_y(j)>(88.8*pi/180)
    angle_factor_pl_y(j)=0;
elseif incid_angle_pl_y(j)<(88.8*pi/180) && incid_angle_pl_y(j)>(50*pi/180)
    angle_factor_pl_y(j)=-0.369*(cos(incid_angle_pl_y(j)))^3+ ...
    ... 0.637*(cos(incid_angle_pl_y(j)))^2+0.750*cos(incid_angle_pl_y(j))-0.015;
else
    angle_factor_pl_y(j)=cos(incid_angle_pl_y(j));
end
power_pl_y(j)=sol_power*area_pl_y(j)*sun_intens(j)*angle_factor_pl_y(j)*0.1977*0.5513;

% minus y panel
if vec(j,18)>0.0676
    area_mi_y(j)=0.0676;
else
    area_mi_y(j)=vec(j,18);
end
effect_area_mi_y(j)=vec(j,19);
incid_angle_mi_y(j)=acos(effect_area_mi_y(j)/area_mi_y(j));
if isnan(incid_angle_mi_y(j))==1
    incid_angle_mi_y(j)=pi/2;
end
if incid_angle_mi_y(j)>(88.8*pi/180)
    angle_factor_mi_y(j)=0;
elseif incid_angle_mi_y(j)<(88.8*pi/180) && incid_angle_mi_y(j)>(50*pi/180)
    angle_factor_mi_y(j)=-0.369*(cos(incid_angle_mi_y(j)))^3+ ...
    ... 0.637*(cos(incid_angle_mi_y(j)))^2+0.750*cos(incid_angle_mi_y(j))-0.015;
else
    angle_factor_mi_y(j)=cos(incid_angle_mi_y(j));
end

```

```

end
power_mi_y(j)=sol_power*area_mi_y(j)*sun_intens(j)*angle_factor_mi_y(j)*0.1977*0.5513;

% total power
power_total(j) = power_up(j)+power_pl_x(j)+power_mi_x(j)+power_pl_y(j)+power_mi_y(j);
power_side(j)   = power_pl_x(j)+power_mi_x(j)+power_pl_y(j)+power_mi_y(j);
en_up           = en_up+power_up(j)*time_step/3600;
en_side         = en_side+power_side(j)*time_step/3600;
en              = en+power_total(j)*time_step/3600;
end
% save calculated energy for this daylight period in energy vectors
energy(i)       = en;
energy_top(i)   = en_up;
energy_side(i)  = en_side;
% progress
progress = i*100/length(duration)
end
% remove solar panel calculation window
root.ExecuteCommand('Window3D * Remove WindowID 1');

save ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\wobble0_amp15,3\ ...
... energy_w0_amp5,3.mat', 'energy', 'energy_*');
Scenario.Unload;
out=1;
end

```

## E.2 MATLAB-Skript zur Auswertung der Daten

### AISat\_mission\_auswertung.m

```

close all
clear all
clc

load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\nadir0\ ...
... lighting_t_nadir0.mat');
time = (start_epmin+stop_epmin)./(2*1440);
dura = duration;

load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\HPOP0\ ...
... lighting_t_HPOP0.mat');
time_HPOP = (start_epmin+stop_epmin)./(2*1440);
dura_HPOP = duration;

% plot duration of lighting periods HPOP vs. J2
figure
plot(time,dura/60,'k',time_HPOP,dura_HPOP/60,'--b','LineWidth',2)
set(gca,'FontName','Palatino','FontSize',12)
legend('J_2','HPOP')
xlabel('Zeit [d]')
ylabel('Dauer [min]')
grid on
%-----
% load energy variables
% perfect nadir, J2
load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\nadir0\ ...
... energy_nadir0.mat');

energy_side0 = energy_side*12/14;
energy_top0  = energy_top*12/14;

load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\nadir45\ ...
... energy_nadir45.mat');

energy_side45 = energy_side*12/14;
energy_top45  = energy_top*12/14;

en_side  = (energy_side45 + energy_side0) ./ 2;
en_top   = (energy_top45  + energy_top0)  ./ 2;
en_total = en_side+en_top;

% perfect nadir HPOP
load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\HPOP0\ ...
... energy_HPOP0.mat');

energy_side0_HPOP = energy_side*12/14;
energy_top0_HPOP  = energy_top*12/14;

load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\HPOP45\ ...
... energy_HPOP45.mat');

energy_side45_HPOP = energy_side*12/14;
energy_top45_HPOP  = energy_top*12/14;

```

```

en_side_HPOP = (energy_side45_HPOP + energy_side0_HPOP)./ 2;
en_top_HPOP = (energy_top45_HPOP + energy_top0_HPOP) ./ 2;
en_total_HPOP = en_side_HPOP+en_top_HPOP;

% wobble, J2
load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\wobble0_amp15,3\ ...
... energy_w0_amp5,3.mat');

energy_side_wob0 = energy_side*12/14;
energy_top_wob0 = energy_top*12/14;

load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\wobble45_amp15,3\ ...
... energy_w45_amp5,3.mat');

energy_side_wob45 = energy_side*12/14;
energy_top_wob45 = energy_top*12/14;

en_side_wob = (energy_side_wob45 + energy_side_wob0)./ 2;
en_top_wob = (energy_top_wob45 + energy_top_wob0) ./ 2;
en_total_wob = en_side_wob + en_top_wob;

load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\wobble0_ampl15,9\ ...
... energy_w0_amp15,9.mat');

energy_side_wob0_ampl15 = energy_side*12/14;
energy_top_wob0_ampl15 = energy_top*12/14;

load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\wobble45_ampl15,9\ ...
... energy_w45_amp15,9.mat');

energy_side_wob45_ampl15 = energy_side*12/14;
energy_top_wob45_ampl15 = energy_top*12/14;

en_side_wob_ampl15 = (energy_side_wob45_ampl15 + energy_side_wob0_ampl15)./ 2;
en_top_wob_ampl15 = (energy_top_wob45_ampl15 + energy_top_wob0_ampl15) ./ 2;
en_total_wob_ampl15 = en_side_wob_ampl15 + en_top_wob_ampl15;

% upside down, perfect zenith, J2
load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\upside_down0\ ...
... energy_up0.mat');

energy_side0_updo = energy_side*12/14;
energy_top0_updo = energy_top*12/14;

load ('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\AISat_Mission\upside_down45\ ...
... energy_up45.mat');

energy_side45_updo = energy_side*12/14;
energy_top45_updo = energy_top*12/14;

en_side_updo = (energy_side45_updo + energy_side0_updo)./ 2;
en_top_updo = (energy_top45_updo + energy_top0_updo) ./ 2;
en_total_updo = en_side_updo+en_top_updo;

%-----
% plot generated energy for HPOP and J2
figure
plot(time,en_side,'k',time_HPOP,en_side_HPOP,'--b',time,en_top,'k',time_HPOP,en_top_HPOP, ...
... '--b',time,en_total,'k',time_HPOP,en_total_HPOP,'--b','LineWidth',2)
set(gca,'FontName','Palatino','FontSize',12)
legend('J_2', 'HPOP','Location',[0.18 0.35 0.1 0.1])
xlabel('Zeit [d]')
xlim([0 370])
ylabel('erzeugte Energie [Wh]')
annotation('textbox',[0.6 0.13 0.1 0.1],'string','oberes Paneel','LineStyle','none', ...
... 'FontName','Palatino','FontSize',12)
annotation('textbox',[0.6 0.4 0.1 0.1],'string','seitliche Paneele','LineStyle','none', ...
... 'FontName','Palatino','FontSize',12)
annotation('textbox',[0.6 0.66 0.1 0.1],'string','gesamt','LineStyle','none', ...
... 'FontName','Palatino','FontSize',12)
grid on

% plot generated energy nad + wob as func of epoch days, J2
figure
plot(time,en_side,'b',time,en_top,'r',time,en_total,'k',time,en_side_wob,'--b',time, ...
... en_top_wob,'--r',time,en_total_wob,'--k',time,en_side_wob_ampl15,'--b',time, ...
... en_top_wob_ampl15,'--r',time,en_total_wob_ampl15,'--k','LineWidth',2)
set(gca,'FontName','Palatino','FontSize',12)
legend('Nadir: seitliche Paneele','Nadir: oberes Paneel','Nadir: gesamt', ...
... 'Schwingungen 5°/3°: seitliche Paneele','Schwingungen 5°/3°: oberes Pannel', ...
... 'Schwingungen 5°/3°: gesamt','Schwingungen 15°/9°: seitliche Paneele', ...
... 'Schwingungen 15°/9°: oberes Paneel','Schwingungen 15°/9°: gesamt', ...
... 'Location','EastOutside')
xlabel('Zeit [d]')
xlim([0 370])
ylabel('erzeugte Energie [Wh]')
grid on

```

```

% plot differences in energy upside down and normal
figure
plot(time,en_side,'--b',time,en_top,'-.b',time,en_total,'b',time,en_side_updo,'--k',time, ...
... en_top_updo,'-.k',time,en_total_updo,'k','LineWidth',2)
set(gca,'FontName','Palatino','FontSize',12)
legend('Nadir: seitliche Paneele','Nadir: oberes Paneel','Nadir: gesamt', ...
... 'Zenit: seitliche Paneele','Zenit: oberes Paneel','Zenit: gesamt', ...
... 'Location','EastOutside')
xlabel('Zeit [d]')
xlim([0 370])
ylabel('erzeugte Energie [Wh]')
grid on
%-----
% calculate and plot battery charge margin, DoD, charge rate
p_reqd = 4.29; % [W], during eclipse
bus_eff = 0.95;
BRF = 1.1;
BCR_eff = 0.9;
capacity = 34.34; % [Wh]
voltage = 7.4; % [V]
cap_Ah = capacity/voltage; % [Ah]

% calculate charge margin for every daylight period of J2 propagation
numlight = length(dura);
tU = 2*pi*sqrt(7028140^3/3.986e14);
E_cha = zeros(numlight,1);
E_cha_wob = zeros(numlight,1);
E_cha_wob_153 = zeros(numlight,1);
E_ill = zeros(numlight,1);
E_ecl = zeros(numlight,1);
E_pan = zeros(numlight,1);
E_pan_wob = zeros(numlight,1);
E_pan_wob_153 = zeros(numlight,1);
margin = zeros(numlight,1);
margin_wob = zeros(numlight,1);
margin_wob_153 = zeros(numlight,1);
DoD = zeros(numlight,1);
I_cha = zeros(numlight,1);
for i = 1:numlight
    E_pan(i) = en_total(i)*3600;
    E_pan_wob(i) = en_total_wob(i)*3600;
    E_pan_wob_153(i) = en_total_wob_ampl15(i)*3600;
    E_ill(i) = p_reqd*dura(i)/(bus_eff*BCR_eff);
    E_ecl(i) = p_reqd*(tU-dura(i))/bus_eff;
    E_cha(i) = (E_pan(i)-E_ill(i))*BCR_eff/BRF;
    E_cha_wob(i) = (E_pan_wob(i)-E_ill(i))*BCR_eff/BRF;
    E_cha_wob_153(i) = (E_pan_wob_153(i)-E_ill(i))*BCR_eff/BRF;
    margin(i) = (E_cha(i)-E_ecl(i))*100/E_cha(i);
    margin_wob(i) = (E_cha_wob(i)-E_ecl(i))*100/E_cha_wob(i);
    margin_wob_153(i) = (E_cha_wob_153(i)-E_ecl(i))*100/E_cha_wob_153(i);
    DoD(i) = E_ecl(i)*100/(3600*capacity);
    % calculate I_cha not with whole sunlight time, but with the fraction
    % of the total sunlight time during which the battery is actually
    % charged, assumed to be equal to 100% - charge margin
    I_cha(i) = (E_ecl(i)*BRF/(voltage*dura(i)*((100-margin(i))/100)*BCR_eff))/cap_Ah;
end

% calculate charge margin for every daylight period of J2 propagation
numlight_HPOP = length(dura_HPOP);
E_cha_HPOP = zeros(numlight_HPOP,1);
E_ill_HPOP = zeros(numlight_HPOP,1);
E_ecl_HPOP = zeros(numlight_HPOP,1);
E_pan_HPOP = zeros(numlight_HPOP,1);
margin_HPOP = zeros(numlight_HPOP,1);
DoD_HPOP = zeros(numlight_HPOP,1);
I_cha_HPOP = zeros(numlight_HPOP,1);
for i = 1:numlight_HPOP
    E_pan_HPOP(i) = en_total_HPOP(i)*3600;
    E_ill_HPOP(i) = p_reqd*dura_HPOP(i)/(bus_eff*BCR_eff);
    E_ecl_HPOP(i) = p_reqd*(tU-dura_HPOP(i))/bus_eff;
    E_cha_HPOP(i) = (E_pan_HPOP(i)-E_ill_HPOP(i))*BCR_eff/BRF;
    margin_HPOP(i) = (E_cha_HPOP(i)-E_ecl_HPOP(i))*100/E_cha_HPOP(i);
    DoD_HPOP(i) = E_ecl_HPOP(i)*100/(3600*capacity);
    I_cha_HPOP(i) = (E_ecl_HPOP(i)*BRF/(voltage*dura_HPOP(i)*...
    ((100-margin_HPOP(i))/100)*BCR_eff))/cap_Ah;
end

figure
plot(time,margin,time,margin_wob,'--',time,margin_wob_153,'-.','LineWidth',2)
set(gca,'FontName','Palatino','FontSize',12)
legend('Nadirausrichtung','Schwingungen 5°/3°','Schwingungen 15°/9°','Location','SouthWest')
xlabel('Zeit [d]')
ylabel('Batteriemargin [%]')
xlim([0 370])
grid on

figure
plot(time,DoD,'k',time_HPOP,DoD_HPOP,'--b','LineWidth',2)
set(gca,'FontName','Palatino','FontSize',12)

```

```

legend('J_2', 'HPOP', 'Location', 'SouthEast')
xlabel('Zeit [d]')
ylabel('DoD [%]')
xlim([0 370])
grid on

figure
plot(time, I_cha, 'k', time_HPOP, I_cha_HPOP, '--b', 'LineWidth', 2)
set(gca, 'FontName', 'Palatino', 'FontSize', 12)
legend('J_2', 'HPOP')
xlabel('Zeit [d]')
ylabel('Laderate [*C_{batt}]')
xlim([0 370])
grid on

%-----
%calculate and plot possible power consumption for upsidedown attitude

P_auf_updo = zeros(numlight,1);
DoD_updo   = zeros(numlight,1);
I_cha_updo = zeros(numlight,1);
for i = 1:numlight
    t_ill = dura(i)/3600; % [h]
    t_ecl = (tU - t_ill)/3600; % [h]
    P_auf_updo(i) = en_total_updo(i)*BCR_eff*bus_eff/(BRF*t_ecl+t_ill); % [W]
    E_ecl = P_auf_updo(i)*t_ecl/bus_eff;
    DoD_updo(i) = E_ecl*100/capacity;
    I_cha_updo(i) = (E_ecl*BRF/(voltage*t_ill*BCR_eff))/cap_Ah; % [*C_{batt}]
end

figure
plot(time, P_auf_updo, 'k', 'LineWidth', 2)
set(gca, 'FontName', 'Palatino', 'FontSize', 12)
xlim([0 370])
xlabel('Zeit [d]')
ylabel('aufnehmbare Leistung [W]')
grid on

%-----
% access

load('D:\STK 9\Results\CLAVIS_Mission\perf_nadir\0deg\lighting_t.mat');
load('D:\STK 9\Results\CLAVIS_Mission\perf_nadir\0deg\access_t.mat');

% total access times in hours
tot_acc_Bre = sum(duration_Bre)/3600;
tot_acc_SNR = sum(time_CN_thresh)/60;

% access time per day
acc_time_per_day = tot_acc_Bre/366
SNR_time_per_day = tot_acc_SNR/366

% plot cumulative access time
cum_acc_Bre = zeros(length(start_time_Bre),1);
cum_acc_SNR = zeros(length(start_time_Bre),1);

for i=1:length(start_time_Bre)
    if i == 1
        cum_acc_Bre(i) = duration_Bre(i)/3600;
        cum_acc_SNR(i) = time_CN_thresh(i)/60;
    else
        cum_acc_Bre(i) = cum_acc_Bre(i-1)+duration_Bre(i)/3600;
        cum_acc_SNR(i) = cum_acc_SNR(i-1)+time_CN_thresh(i)/60;
    end
end

figure
plot(stop_time_Bre/1440, cum_acc_Bre, '--b', stop_time_Bre/1440, cum_acc_SNR, 'k', 'LineWidth', 2)
set(gca, 'FontName', 'Palatino', 'FontSize', 12)
xlim([0 370])
xlabel('Zeit [d]')
ylabel('kumulierte Bodenkontaktzeit [h]')
legend('Bodenkontaktzeit', 'Bodenkontaktzeit mit ausreichendem CNR', 'Location', 'NorthWest')
grid on

% calculate cumulated amount of data
connect_time = 2/60; % time loss for establishing connection, [min]
RS_code_loss = 223/255; % data loss through (255,223)-Reed-Solomon-Code
margin = 2; % loss factor due to night time etc
data_rate = 4800*60/(8*1024*1024); % MB per minute
F_red = RS_code_loss/margin;
cum_data = zeros(length(start_time_Bre),1);
for i = 1:length(start_time_Bre)
    if i == 1
        cum_data(i) = (time_CN_thresh(i)-connect_time)*data_rate*F_red;
    else
        if time_CN_thresh < connect_time
            cum_data(i) = cum_data(i-1);
        else

```



```

        cum_data(i) = cum_data(i-1)+(time_CN_thresh(i)-connect_time)*data_rate*F_red;
    end
end
figure
plot(stop_time_Bre/1440,cum_data,'LineWidth',2)
xlabel('epoch days')
ylabel('cumulated amount of data sent [MB]')
grid on

cum_data_per_day = cum_data(length(cum_data))/366

```

## E.3 Skripte zur Simulation der Verläufe über einen Orbit

### E.3.1 Skript zur Simulation

Das hier wiedergegebene Skript wird zur Simulation des Verlaufs der von der AISat-Konfiguration während eines Orbits erzeugten Energie verwendet. Ähnliche Skripte existieren für die anderen Konfigurationen zum Vergleich der verschiedenen Konfigurationen und finden sich auf der CD zur Arbeit. Sie unterscheiden sich lediglich in der Definition des verwendeten Satellitenmodells.

#### AISat\_compute\_one\_orbit\_power.m

```

clear all
close all
clc

%-----
% Hook to running STK
%-----
uiApp = actxGetRunningServer('STK9.application');
root = uiApp.Personality2;

% define units
root.UnitPreferences.SetCurrentUnit('DateFormat', 'UTCG');
root.UnitPreferences.SetCurrentUnit('DistanceUnit', 'km');
root.UnitPreferences.SetCurrentUnit('TimeUnit', 'sec');
root.UnitPreferences.SetCurrentUnit('AngleUnit', 'deg');

%-----
% Create new objects and set their properties
%-----
% create new scenario, define scenario object
root.NewScenario('CLAVIS_Matlab_neu');
Scenario = root.GetObjectFromPath('*/');
Scenario.StopTime = '1 Mar 2012 01:38:00.0';
Scenario.StartTime = '1 Mar 2012 00:00:00.0';

root.UnitPreferences.SetCurrentUnit('DateFormat', 'ModJDate');

% maximize and hide window
root.ExecuteCommand('Application / Maximize');
root.ExecuteCommand('Application / Hide');

% create Satellite CLAVIS
Satellite = Scenario.Children.New('eSatellite', 'CLAVIS');
Satellite.SetPropagatorType('ePropagatorJ4Perturbation');
Satellite.Propagator.UseScenarioAnalysisTime;
Satellite.Propagator.InitialState.Representation.AssignClassical ...
    ... ('eCoordinateSystemJ2000', 7028.14, 0.0, 97.998, 0.0, 129.203, 0.0);
Satellite.Propagator.Propagate();

% set attitude and model file for satellite
Satellite.SetAttitudeType('eAttitudeStandard');
Satellite.Attitude.Basic.SetProfileType('eProfileSpinAboutNadir');
Satellite.Attitude.Basic.Profile.Epoch = 55805;
Satellite.Attitude.Basic.Profile.Rate = 0;
Satellite.Attitude.Basic.Profile.Offset = 22.5;

Satellite.VO.Model.ModelType='eModelFile';
Satellite.VO.Model.ModelData.FileName='STKData\VO\Models\Space\CLAVIS_AISat_full.mdl';

%-----
% compute solar panel exposure data
%-----

```

```

time_step=1; %s
% set distance unit in meters -> solar panel area in m^2
root.UnitPreferences.SetCurrentUnit('DistanceUnit', 'm');
root.UnitPreferences.SetCurrentUnit('DateFormat', 'EpMin');
% Add all Solar Panel groups
root.ExecuteCommand('VO */Satellite/CLAVIS SolarPanel Visualization AddGroup CLAVIS_up ...
... AddGroup CLAVIS_+x AddGroup CLAVIS_-x AddGroup CLAVIS_+y AddGroup CLAVIS_-y ...
... View On Radius On 0.5');
% set offscreen rendering on, view must be on in VO SolarPanel Visualisation
root.ExecuteCommand('Window3D * SetRenderMethod Method PBuffer WindowID 2');
root.ExecuteCommand('Window3D * Maximize WindowID 2');
display('computing solar panel exposure...')
% actually compute data
root.ExecuteCommand('VO */Satellite/CLAVIS SolarPanel Compute "1 Mar 2012 00:00:00.0" ...
... "1 Mar 2012 01:38:00.0" 1 Area "D:\area_report.txt" ');
% save report output
DataProvider = Satellite.DataProviders.Item('Solar Panel Area');
ReportElements = {'Time'; 'Area'; 'Effective Area'; 'Solar Intensity'};
Results = DataProvider.ExecElements(0, 98, time_step, ReportElements);
Data = Results.DataSets;
% delete Solar panel data
root.ExecuteCommand('VO */Satellite/CLAVIS SolarPanel DeleteData');
% remove solar panel calculation window
root.ExecuteCommand('Window3D * Remove WindowID 2');

%-----
% evaluate data
%-----

display('save data in matrix')
vec_cell = Data.ToArray;
le = length(Data.GetRow(1));
vec = zeros(Data.RowCount,le);
for j = 1:Data.RowCount
    for i = 1:le
        vec(j,i) = vec_cell{j,i};
    end
end

display('preallocate vectors')
energy=0;
energy_top=0;
% save data in vectors
% general
time=zeros(Data.RowCount-1,1);
sun_intens=zeros(Data.RowCount-1,1);
power_total=zeros(Data.RowCount-1,1);

% top panel
area_up=zeros(Data.RowCount-1,1);
effect_area_up=zeros(Data.RowCount-1,1);
incid_angle_up=zeros(Data.RowCount-1,1);
angle_factor_up=zeros(Data.RowCount-1,1);
power_up=zeros(Data.RowCount-1,1);

% + x panel
area_pl_x=zeros(Data.RowCount-1,1);
effect_area_pl_x=zeros(Data.RowCount-1,1);
incid_angle_pl_x=zeros(Data.RowCount-1,1);
angle_factor_pl_x=zeros(Data.RowCount-1,1);
power_pl_x=zeros(Data.RowCount-1,1);

% - x panel
area_mi_x=zeros(Data.RowCount-1,1);
effect_area_mi_x=zeros(Data.RowCount-1,1);
incid_angle_mi_x=zeros(Data.RowCount-1,1);
angle_factor_mi_x=zeros(Data.RowCount-1,1);
power_mi_x=zeros(Data.RowCount-1,1);

% + y panel
area_pl_y=zeros(Data.RowCount-1,1);
effect_area_pl_y=zeros(Data.RowCount-1,1);
incid_angle_pl_y=zeros(Data.RowCount-1,1);
angle_factor_pl_y=zeros(Data.RowCount-1,1);
power_pl_y=zeros(Data.RowCount-1,1);

% - y panel
area_mi_y=zeros(Data.RowCount-1,1);
effect_area_mi_y=zeros(Data.RowCount-1,1);
incid_angle_mi_y=zeros(Data.RowCount-1,1);
angle_factor_mi_y=zeros(Data.RowCount-1,1);
power_mi_y=zeros(Data.RowCount-1,1);

display('start loop to compute power')
%loop over every time step
for i=1:(Data.RowCount-1)
    % general data
    time(i)=vec(i,1);

```

```

sun_intens(i)=vec(i,4);

% top panel
if vec(i,2)>0.0676
    area_up(i)=0.0676;
else
    area_up(i)=vec(i,2);
end
effect_area_up(i)=vec(i,3);
incid_angle_up(i)=acos(effect_area_up(i)/area_up(i));
if isnan(incid_angle_up(i))==1
    incid_angle_up(i)=pi/2;
end
if incid_angle_up(i)>(88.8*pi/180)
    angle_factor_up(i)=0;
elseif incid_angle_up(i)<(88.8*pi/180) && incid_angle_up(i)>(50*pi/180)
    angle_factor_up(i)=-0.369*(cos(incid_angle_up(i)))^3+ ...
        ... 0.637*(cos(incid_angle_up(i)))^2+0.750*cos(incid_angle_up(i))-0.015;
else
    angle_factor_up(i)=cos(incid_angle_up(i));
end
power_up(i)=1325.8*area_up(i)*sun_intens(i)*angle_factor_up(i)*0.1977*0.5513;
energy_top=energy_top+power_up(i)/60;

% plus x panel
if vec(i,6)>0.0676
    area_pl_x(i)=0.0676;
else
    area_pl_x(i)=vec(i,6);
end
effect_area_pl_x(i)=vec(i,7);
incid_angle_pl_x(i)=acos(effect_area_pl_x(i)/area_pl_x(i));
if isnan(incid_angle_pl_x(i))==1
    incid_angle_pl_x(i)=pi/2;
end
if incid_angle_pl_x(i)>(88.8*pi/180)
    angle_factor_pl_x(i)=0;
elseif incid_angle_pl_x(i)<(88.8*pi/180) && incid_angle_pl_x(i)>(50*pi/180)
    angle_factor_pl_x(i)=-0.369*(cos(incid_angle_pl_x(i)))^3+ ...
        ... 0.637*(cos(incid_angle_pl_x(i)))^2+0.750*cos(incid_angle_pl_x(i))-0.015;
else
    angle_factor_pl_x(i)=cos(incid_angle_pl_x(i));
end
power_pl_x(i)=1325.8*area_pl_x(i)*sun_intens(i)*angle_factor_pl_x(i)*0.1977*0.5513;

% minus x panel
if vec(i,10)>0.0676
    area_mi_x(i)=0.0676;
else
    area_mi_x(i)=vec(i,10);
end
effect_area_mi_x(i)=vec(i,11);
incid_angle_mi_x(i)=acos(effect_area_mi_x(i)/area_mi_x(i));
if isnan(incid_angle_mi_x(i))==1
    incid_angle_mi_x(i)=pi/2;
end
if incid_angle_mi_x(i)>(88.8*pi/180)
    angle_factor_mi_x(i)=0;
elseif incid_angle_mi_x(i)<(88.8*pi/180) && incid_angle_mi_x(i)>(50*pi/180)
    angle_factor_mi_x(i)=-0.369*(cos(incid_angle_mi_x(i)))^3+ ...
        ... 0.637*(cos(incid_angle_mi_x(i)))^2+0.750*cos(incid_angle_mi_x(i))-0.015;
else
    angle_factor_mi_x(i)=cos(incid_angle_mi_x(i));
end
power_mi_x(i)=1325.8*area_mi_x(i)*sun_intens(i)*angle_factor_mi_x(i)*0.1977*0.5513;

% plus y panel
if vec(i,14)>0.0676
    area_pl_y(i)=0.0676;
else
    area_pl_y(i)=vec(i,14);
end
effect_area_pl_y(i)=vec(i,15);
incid_angle_pl_y(i)=acos(effect_area_pl_y(i)/area_pl_y(i));
if isnan(incid_angle_pl_y(i))==1
    incid_angle_pl_y(i)=pi/2;
end
if incid_angle_pl_y(i)>(88.8*pi/180)
    angle_factor_pl_y(i)=0;
elseif incid_angle_pl_y(i)<(88.8*pi/180) && incid_angle_pl_y(i)>(50*pi/180)
    angle_factor_pl_y(i)=-0.369*(cos(incid_angle_pl_y(i)))^3+ ...
        ... 0.637*(cos(incid_angle_pl_y(i)))^2+0.750*cos(incid_angle_pl_y(i))-0.015;
else
    angle_factor_pl_y(i)=cos(incid_angle_pl_y(i));
end
power_pl_y(i)=1325.8*area_pl_y(i)*sun_intens(i)*angle_factor_pl_y(i)*0.1977*0.5513;

% minus y panel
if vec(i,18)>0.0676

```

```

        area_mi_y(i)=0.0676;
    else
        area_mi_y(i)=vec(i,18);
    end
    effect_area_mi_y(i)=vec(i,19);
    incid_angle_mi_y(i)=acos(effect_area_mi_y(i)/area_mi_y(i));
    if isnan(incid_angle_mi_y(i))==1
        incid_angle_mi_y(i)=pi/2;
    end
    if incid_angle_mi_y(i)>(88.8*pi/180)
        angle_factor_mi_y(i)=0;
    elseif incid_angle_mi_y(i)<(88.8*pi/180) && incid_angle_mi_y(i)>(50*pi/180)
        angle_factor_mi_y(i)=-0.369*(cos(incid_angle_mi_y(i)))^3+ ...
        ... 0.637*(cos(incid_angle_mi_y(i)))^2+0.750*cos(incid_angle_mi_y(i))-0.015;
    else
        angle_factor_mi_y(i)=cos(incid_angle_mi_y(i));
    end
    power_mi_y(i)=1325.8*area_mi_y(i)*sun_intens(i)*angle_factor_mi_y(i)*0.1977*0.5513;

    % total power
    power_total(i)=power_up(i)+power_pl_x(i)+power_mi_x(i)+power_pl_y(i)+power_mi_y(i);
    energy=energy+power_total(i)*time_step/3600;

    % progress
    progress=i*100/(Data.RowCount-1)
end

Scenario.Unload;

save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\one_orbit\AISat\power_AISat.mat', ...
    ... 'power_*');
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\one_orbit\AISat\angles_AISat.mat', ...
    ... 'incid_angle_*');
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\one_orbit\AISat\time_AISat.mat', ...
    ... 'time');

```

## E.3.2 Auswertungsskript

### AISat\_one\_orbit\_auswertung.m

```

clear all
close all
clc

load('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\one_orbit\AISat\power_AISat.mat');
load('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\one_orbit\AISat\angles_AISat.mat');
load('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\one_orbit\AISat\time_AISat.mat');

% plot power
plot(time,power_total.*12./14,'--k',time,power_up.*12./14,'k',time,power_pl_x.*12./14,':b', ...
    ... time,power_mi_x.*12./14,'--b',time,power_pl_y.*12./14,':r',time,power_mi_y.*12./14, ...
    ... '--r','LineWidth',2)
set(gca,'FontName','Palatino','FontSize',12)
legend('gesamt','oberes Paneel','+x Paneel','-x Paneel','+y Paneel','-y Paneel', ...
    ... 'Location','EastOutside');
xlabel('Zeit [min]')
ylabel('Leistung [W]')
grid on

% plot angles
figure
plot(time,incid_angle_up*180/pi,'k',time,incid_angle_pl_x*180/pi,':b', ...
    ... time,incid_angle_mi_x*180/pi,'--b',time,incid_angle_pl_y*180/pi,':r', ...
    ... time,incid_angle_mi_y*180/pi,'--r','LineWidth',2)
set(gca,'FontName','Palatino','FontSize',12)
legend('oberes Paneel','+x Paneel','-x Paneel','+y Paneel','-y Paneel', ...
    ... 'Location','EastOutside');
xlabel('Zeit [min]')
ylabel('Einstrahlwinkel [°]')
grid on

% battery calculations
lt = length(time);
for i = 1:lt
    if i ~= 1
        if power_total(i) ~= 0 && power_total(i-1) == 0
            sun_start_time = time(i); % first power value ~= 0
            sun_start_ind = i;
        elseif power_total(i) == 0 && power_total(i-1) ~= 0
            sun_stop_time = time(i-1); % last power value ~= 0
            sun_stop_ind = i-1;
        end
    end
end
end

```

```

% in seconds:
tU      = 2*pi*sqrt(7028140^3/3.986e14);
sun_dur = (sun_stop_time-sun_start_time)*60;
ecl_dur = tU-sun_dur;

batt_cap_Wh = zeros(1t,1);
rate        = zeros(1t,1);
amperage     = zeros(1t,1);

% battery constants
p_reqd      = 4.29;    % [W], during eclipse
bus_eff     = 0.95;
BRF         = 1.1;
BCR_eff     = 0.9;
capacity    = 34.34;   % [Wh]
capacity_Ah = 4.64;    % [Ah]
voltage     = 7.4;     % [V]
% energy per second
en_reqd_sec_Wh = p_reqd/(bus_eff*3600);    % [Wh]
% energy required in eclipse
en_reqd_ecl_Wh = p_reqd*ecl_dur/(3600*bus_eff);
% capacity of battery @ sun_start_time
cap_bat_sun_start_Wh = capacity-en_reqd_ecl_Wh;
% capacity of battery @ t=1s
batt_cap_Wh(1) = cap_bat_sun_start_Wh+(sun_start_time*60-2)*en_reqd_sec_Wh;
% batt charge min
batt_charge_min_Wh = cap_bat_sun_start_Wh;
for i = 2:1t
    if time(i) < sun_start_time
        batt_cap_Wh(i) = batt_cap_Wh(i-1)-en_reqd_sec_Wh;
    elseif (time(i) >= sun_start_time) && (time(i) <= sun_stop_time)
        power_avail = power_total(i)*BCR_eff*bus_eff*12/14;
        energ_avail = power_avail/3600;    % Wh per second
        % energy from panels not enough to supply power to instruments:
        if energ_avail < en_reqd_sec_Wh
            % energy supplied by battery:
            en_from_batt_Wh = (en_reqd_sec_Wh-energ_avail)/bus_eff;
            % new battery capacity
            batt_cap_Wh(i) = batt_cap_Wh(i-1)-en_from_batt_Wh;
            if batt_cap_Wh(i) < batt_charge_min_Wh
                batt_charge_min_Wh = batt_cap_Wh(i);
            end
        else
            % maximum charge level
            max_charge_Wh = batt_charge_min_Wh+(capacity-batt_charge_min_Wh)*BRF;
            % max charge level reached?
            if batt_cap_Wh(i-1) > max_charge_Wh
                batt_cap_Wh(i) = capacity;
            elseif batt_cap_Wh(i-1) == capacity
                batt_cap_Wh(i) = capacity;
            else
                batt_cap_Wh(i) = batt_cap_Wh(i-1)+energ_avail-en_reqd_sec_Wh;
            end
        end
    else
        batt_cap_Wh(i) = batt_cap_Wh(i-1)-en_reqd_sec_Wh;
    end
    % compute charge rate
    amperage(i) = (batt_cap_Wh(i)-batt_cap_Wh(i-1))*3600/voltage;
    rate(i)     = amperage(i)/capacity_Ah;
end

rate(1)      = rate(2);
% vectors for sunlight start and stop tokens
time_start = [sun_start_time sun_start_time];
time_stop  = [sun_stop_time sun_stop_time];
perc       = [92 101];
chara      = [-0.2 0.3];

figure
plot(time,batt_cap_Wh*100/34.34,'LineWidth',2)
set(gca,'FontName','Palatino','FontSize',12)
xlabel('Zeit [min]')
ylabel('Batteriekapazität [%]')
grid on
hold on
plot(time_start,perc,'k',time_stop,perc,'k')
hold off

```

## Anhang F

# Simulation der zukünftigen CLAVIS-Missionen

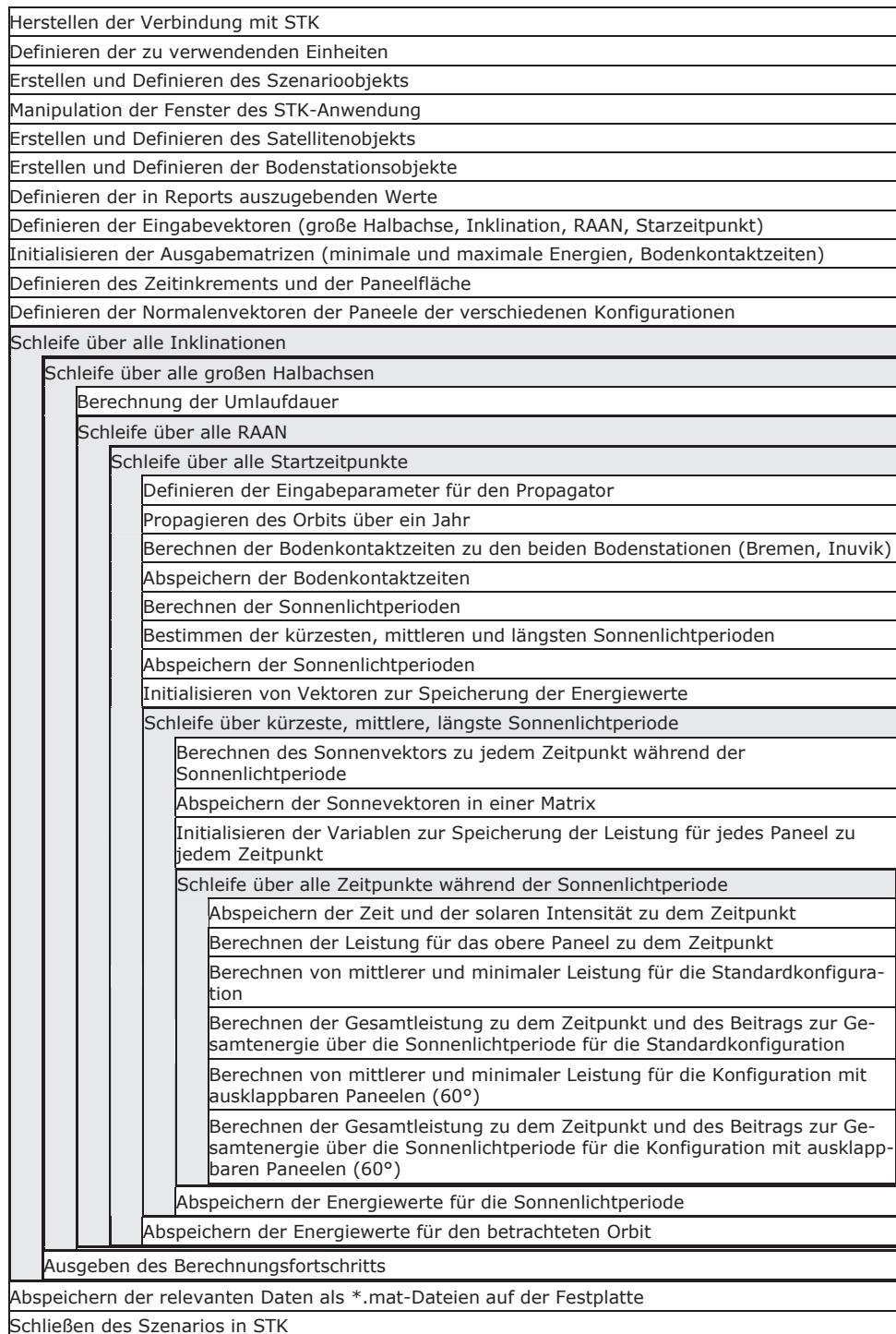
In diesem Abschnitt wird zunächst das MATLAB-Skript zur Simulation der zukünftigen CLAVIS-Missionen detailliert beschrieben und im Anschluss wiedergegeben, mit dem die Richtwerte für die erzeugte Energie und die Sonnenlichtdauer als Funktion der Bahnelemente bestimmt werden (siehe Abschnitte 10.2 und 10.4). Das Skript zur Bestimmung der erzeugten Energie als Funktion der Sonnenlichtdauer ist dem sehr ähnlich und wird ebenfalls wiedergegeben (siehe Abschnitt 10.3). Anschließend werden beispielhaft Skripte zur Auswertung der gewonnenen Daten gezeigt. Abschließend wird ein Skript zur Simulation der Evolution der Bahnhöhe wiedergegeben.

### F.1 Skript zur Simulation zukünftiger CLAVIS-Missionen

#### F.1.1 Skript zur Berechnung der Sonnenlichtzeiten und der erzeugten Energie als Funktion der Bahnparameter

Abbildung F.1 zeigt ein vereinfachtes Struktogramm des MATLAB-Skripts zur Berechnung der Sonnenlichtzeiten und der erzeugten Energie als Funktion der Bahnparameter für mögliche zukünftige CLAVIS-Missionen. Wie schon bei dem Skript für die AISat-Mission steht auch hier ein Anweisungsblock für mehrere Codezeilen.

Der Beginn des Skripts ähnelt dem in Abschnitt E.1 beschriebenen, wobei das Szenarioobjekt hier über zwei Jahre definiert wird. Damit können verschiedene Startzeitpunkte während des ersten Jahres betrachtet werden, ohne dass das Ende der Mission außerhalb des Szenariozeitraumes liegt. Die Definition eines einzigen Szenarios mit verschiedenen Missionszeiträumen ist deutlich recheneffektiver als die Neudefinition des Szenarioobjekts für jeden betrachteten Startzeitpunkt. Die Bahn des Satelliten wird in diesem Fall auch nicht zu Beginn des Skripts propagiert. Dies geschieht erst in den Schleifen über die verschiedenen variierten Parameter. Darüber hinaus ist hier die Definition des Sender- und Empfängerobjekts nicht nötig, da das Linkbudget nicht betrachtet wird (siehe auch Abschnitt 8.1.2).



**Abbildung F.1:** Vereinfachtes Struktogramm eines MATLAB-Skripts zur Berechnung der Rohdaten zur Analyse der zukünftigen CLAVIS-Missionen

Nach den grundlegenden Definitionen werden die in den Reports auszugebenden Daten sowie die Vektoren, die die Eingabeparameter, also die verschiedenen Startzeitpunkte, RAAN, Inklinationen und großen Halbachsen der betrachteten Orbits enthalten, definiert. Als Datumsangabe wird dabei der Einfachheit halber stets das Julianische Datum verwendet. Daraufhin werden die Datenstrukturen zur Speiche-

rung der Sonnenlichtzeiten, Energien und Bodenkontaktzeiten initialisiert, die von allen Eingabeparametern abhängen, damit also vierdimensional sind. Außerdem werden die in Abschnitt 8.3.2 gegebenen Normalenvektoren der Solarpaneele der verschiedenen Konfigurationen definiert.

Daraufhin werden vier *for*-Schleifen begonnen, die jeweils über die Länge der vier Eingabevektoren (Startzeitpunkt, RAAN, Inklination, große Halbachse) laufen. Die Reihenfolge ist dabei willkürlich gewählt, wobei die Berechnung der Umlaufdauer erfolgen kann, sobald die Schleife über die großen Halbachsen begonnen wurde. Innerhalb dieser Schleifen wird zunächst dem Propagator ein Start- und ein Endzeitpunkt sowie die Epoche des Orbits zugewiesen. Start- und Endzeitpunkt ergeben sich aus dem entsprechenden Eingabevektor. Die Epoche des Orbits darf hier nicht ausgelassen werden, da ansonsten standardmäßig der Beginn des Szenarios als Epoche des Orbits angenommen wird, was dazu führt, dass immer der gleiche Startorbit nur über verschiedene Zeiträume propagiert wird. Anschließend wird dem  $J_2$ -Propagator der durch die Eingabevektoren definierte Orbit übergeben und propagiert.

Damit ist die Bahn des Satelliten für die gesamte Mission bekannt und die Berechnung der interessierenden Größen kann beginnen. Zunächst werden die Bodenkontaktzeiten zu den Bodenstationen in Bremen und Inuvik berechnet und jeweils die Gesamtdauer über die Mission, die Anzahl der Kontaktperioden und die maximale Dauer einer Kontaktperiode gespeichert.

Anschließend werden die Sonnenlichtzeiten berechnet. Aus den Daten wird dann jeweils die kürzeste und die längste Sonnenlichtperiode sowie die längste mit einer Dauer niedriger als die Umlaufdauer des Satelliten bestimmt. Außerdem wird diejenige Sonnenlichtperiode gesucht, deren Dauer dem Mittelwert aller Sonnenlichtperioden, die kürzer als die Umlaufzeit des Satelliten sind, am nächsten kommt. Dies hat den Hintergrund, dass Sonnenlichtperioden in dem hier betrachteten LEO-Bereich meistens entweder kürzer als die Umlaufdauer oder deutlich länger (einige Tage) sind. Die besonders langen Perioden sind aus der Sicht der Energieerzeugung nicht weiter kritisch. Um einen konservativen Wert für die durchschnittliche Dauer des Großteils der kürzeren Sonnenlichtperioden zu erhalten, werden die sehr langen bei der Berechnung vernachlässigt. Anfangs-, Endzeitpunkt und Dauer der kürzesten und mittleren Sonnenlichtperioden sowie der längsten mit einer kürzeren Dauer als der Umlaufzeit werden in entsprechenden Vektoren gespeichert. In der folgenden Darstellung wird die letztere der Einfachheit halber als die „längste“ Sonnenlichtperiode bezeichnet.

Anschließend werden Vektoren zur Speicherung der Energie für die drei Fälle initialisiert und eine weitere *for*-Schleife über die drei Fälle wird begonnen. Innerhalb dieser Schleife werden für den jeweils betrachteten Fall zunächst die Einheitsvektoren vom Satelliten zur Sonne im körperfesten Koordinatensystem (Sonneneinheitsvektoren) in 60-Sekunden-Schritten während der betrachteten Sonnenlichtperiode berechnet. Dazu werden der *ExecElements*-Methode Start- und Endzeitpunkt der Sonnenlichtperiode sowie das Zeitinkrement übergeben. Die Vektoren werden wiederum in einer Matrix gespeichert.



Darauf folgt ähnlich dem Skript zur Simulation der AISat-Mission die Definition von Vektoren zur Speicherung der relevanten Werte zur Bestimmung der zu einem bestimmten Zeitpunkt erzeugten Leistung und der über eine Sonnenlichtperiode erzeugten Energie. Es schließt sich eine weitere *for*-Schleife über alle Zeitpunkte während der aktuell betrachteten Sonnenlichtperiode an. In dieser werden ebenfalls zunächst der Zeitpunkt sowie die Solar Intensity in dafür vorgesehenen Vektoren gespeichert und daraufhin die vom oberen Paneel erzeugte Leistung zu dem gegebenen Zeitpunkt über den Winkel zwischen dem Sonneneinheitsvektor und dem Normalenvektor des Paneels berechnet (siehe Abschnitt 8.3.2).

Zur Berechnung der durch die seitlichen Paneele erzeugten Energie müssen die Ergebnisse aus Abschnitt 10.3.1 berücksichtigt werden. Die über eine Sonnenlichtperiode erzeugte Energie hängt direkt mit der Dauer der Sonnenlichtperiode zusammen. Für die Standardkonfiguration bedeutet dies, dass die erzeugte Energie mit der Dauer der Sonnenlichtperiode steigt, für die Konfiguration mit ausklappbaren Paneelen gilt gerade das Gegenteil. Daher wird die von den seitlichen Paneelen, die den Unterschied zwischen den Konfigurationen ausmachen, erzeugte Energie nur für die relevanten Fälle berechnet. Dies bedeutet, dass für die Standardkonfiguration nur dann die von den seitlichen Paneelen erzeugte Energie berechnet wird, wenn gerade die Sonnenlichtperiode kurzer oder mittlerer Dauer betrachtet wird und für die Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen nur die Sonnenlichtperioden mittlerer und maximaler Dauer betrachtet werden. Damit werden für beide Konfigurationen ein durchschnittlicher und ein schlechter Fall untersucht. Die Unterscheidung zwischen den Fällen läuft über eine *if*-Abfrage der Zählvariable der *for*-Schleife über die drei Fälle der Sonnenlichtperioden.

Zur Bestimmung der von den seitlichen Paneelen erzeugten Energie unter Berücksichtigung der Rotation des Satelliten wird die von den vier Paneelen mit einem Drehwinkel von  $0^\circ$  und von den vier Paneelen mit einem Drehwinkel von  $45^\circ$  erzeugte Energie aufsummiert und anschließend halbiert (siehe auch Abschnitt 8.3.2). Damit ist ein durchschnittlicher Wert der gesamten von den seitlichen Paneelen erzeugten Energie berechnet. Die einzelnen Werte werden in entsprechenden Vektoren abgespeichert. Dieser Vorgang wird für die Standardkonfiguration und die Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen durchgeführt. Für die Konfiguration mit um  $90^\circ$  ausgeklappten Paneelen kann die von den seitlichen Paneelen erzeugte Energie aus der vom oberen Paneel erzeugten Energie berechnet werden.

Damit endet die Schleife über alle Zeitpunkte und die aufsummierten Energiewerte über die Sonnenlichtperiode werden in entsprechende Vektoren eingelesen. Damit endet auch die Schleife über die drei Fälle der Dauer der Sonnenlichtperioden und die darin erzeugten Werte werden in den endgültigen Datenstrukturen zur Speicherung auf dem lokalen Laufwerk eingelesen. Nach der Ausgabe des Rechnungsfortschritts enden auch die Schleifen über die vier Eingangsgrößen.

Abschließend werden die relevanten Daten auf der Festplatte gespeichert. Dies sind die minimalen und durchschnittlichen Werte der Energie für das obere Paneel sowie die seitlichen Paneele in den beiden betrachteten Konfigurationen, die jeweils kürzeste,

längste und durchschnittliche Sonnenlichtdauer, die Eingabevektoren und die genannten Kennwerte der Bodenkontaktzeiten. Das Szenario in STK wird daraufhin geschlossen. Die entsprechenden Ausgabedateien finden sich auf der CD zur Arbeit.

## Quellcode

Das hier gezeigte Skript ist das oben beschriebene zur Berechnung der Sonnenlichtzeiten und der erzeugten Energie als Funktion der Bahnparameter auf Orbits mit Inklinationen zwischen  $0^\circ$  und  $90^\circ$ . Für SSOs existiert ein sehr ähnliches Skript, das sich lediglich in den Eingangsgrößen unterscheidet und hier daher nicht wiedergegeben wird. Es befindet sich auf der CD zur Arbeit. Auch dabei werden zu lange Codezeilen unter Verwendung von drei aufeinander folgenden Punkten umgebrochen (siehe Abschnitt C.2.4).

### compute\_power\_access\_CLAVIS.m

```
function [out]=compute_power_access_CLAVIS()

%-----
display('Hook to running STK...')
%-----
uiApp    = actxGetRunningServer('STK9.application');
root     = uiApp.Personality2;

% define units
root.UnitPreferences.SetCurrentUnit('DateFormat', 'JDate');
root.UnitPreferences.SetCurrentUnit('DistanceUnit', 'm');
root.UnitPreferences.SetCurrentUnit('TimeUnit', 'sec');
root.UnitPreferences.SetCurrentUnit('AngleUnit', 'deg');

%-----
display('Create new objects and set their properties...')
%-----
% create new scenario, define scenario object
root.NewScenario('CLAVIS_Matlab_neu');
Scenario = root.GetObjectFromPath('*/');
jd_start = jday(2012,01,01,00,00,00);
jd_stop  = jday(2014,01,01,00,00,00);
Scenario.StopTime = jd_stop;
Scenario.StartTime = jd_start;

% use standard terrain data file
Scenario.TerrainCollection.Item(0).UseTerrain = 1;

% remove standard 2D and 3D windows
root.ExecuteCommand('Window3D * Remove WindowID 1');
root.ExecuteCommand('Window2D * Remove WindowID 1');

% maximize and hide main window
root.ExecuteCommand('Application / Maximize');
root.ExecuteCommand('Application / Hide');

% create Satellite CLAVIS
Satellite = Scenario.Children.New('eSatellite', 'CLAVIS');
Satellite.SetPropagatorType('ePropagatorJ2Perturbation');

% set attitude and model file for satellite
Satellite.SetAttitudeType('eAttitudeStandard');
Satellite.Attitude.Basic.SetProfileType('eProfileSpinAboutNadir');
Satellite.Attitude.Basic.Profile.Rate = 0;
Satellite.Attitude.Basic.Profile.Offset = 0;

Satellite.VO.Model.ModelType = 'eModelFile';
Satellite.VO.Model.ModelData.FileName = 'STKData\VO\Models\Space\CLAVIS_Standard.mdl';

% only Earth as eclipse body
if Satellite.EclipseBodies.IsCentralBodyAssigned('Moon') == 1
    Satellite.EclipseBodies.UseCustomizedList = 1;
    Satellite.EclipseBodies.AssignCentralBody('Earth');
end

% create new facility Bremen
Bremen = Scenario.Children.New('eFacility', 'Bremen');
Bremen.Position.AssignGeodetic(53.111, 8.862, 0);
Bremen.UseTerrain = 1;
% constraints for facility
```

```

Elev_Constraint_Bre = Bremen.AccessConstraints.AddConstraint('eCstrElevationAngle');
Elev_Constraint_Bre.EnableMin = 1;
Elev_Constraint_Bre.Min = 10;

% create new facility Inuvik
Inuvik = Scenario.Children.New('eFacility','Inuvik');
Inuvik.Position.AssignGeodetic(68.318, -133.545, 0);
Inuvik.UseTerrain = 1;
% constraints for facility
Elev_Constraint_Inu = Inuvik.AccessConstraints.AddConstraint('eCstrElevationAngle');
Elev_Constraint_Inu.EnableMin = 1;
Elev_Constraint_Inu.Min = 10;

%-----
display('prepare outputs/report...')
%-----
% report content for lighting times
DataProvider = Satellite.DataProviders.Item('Lighting Times').Group.Item('Sunlight');
RptElems = {'Start Time'; 'Stop Time'; 'Duration'};

% report content for access
RptElems_acc = {'Duration'};

% report content for power calc, unit sun vector in body frame
DataProvider_power = Satellite.DataProviders.Item('Vectors(Body)').Group.Item('Sun');
ReportElements_power = {'Time'; 'x/Magnitude'; 'y/Magnitude'; 'z/Magnitude'};

% input start time
sat_start = jday(2012,01,01,00,00,00);
sat_start_time = sat_start:21:(sat_start+365);
sat_stop_time = sat_start_time+365.25;
numtimes = length(sat_start_time);

% input RAAN
RAAN_deg = 0:20:360;
numRAAN = length(RAAN_deg);

% input altitude
sma = 6578140:50000:7028140;
num sma = length(sma);

% input inclination
incl = 0:5:90;
numincl = length(incl);

% total number
number = numtimes*numRAAN*num sma*numincl

% output vectors power
min_energy = zeros(numtimes,numRAAN,num sma,numincl);
ave_energy = zeros(numtimes,numRAAN,num sma,numincl);
min_energy_up = zeros(numtimes,numRAAN,num sma,numincl);
ave_energy_up = zeros(numtimes,numRAAN,num sma,numincl);
min_energy_side = zeros(numtimes,numRAAN,num sma,numincl);
ave_energy_side = zeros(numtimes,numRAAN,num sma,numincl);
min_duration = zeros(numtimes,numRAAN,num sma,numincl);
ave_duration = zeros(numtimes,numRAAN,num sma,numincl);
max_duration = zeros(numtimes,numRAAN,num sma,numincl);
max_duration_sm_tu = zeros(numtimes,numRAAN,num sma,numincl);
min_energy_depl = zeros(numtimes,numRAAN,num sma,numincl);
ave_energy_depl = zeros(numtimes,numRAAN,num sma,numincl);
min_energy_side_depl = zeros(numtimes,numRAAN,num sma,numincl);
ave_energy_side_depl = zeros(numtimes,numRAAN,num sma,numincl);

% output vectors access
acc_dur_Bre = zeros(numtimes,numRAAN,num sma,numincl);
acc_dur_Inu = zeros(numtimes,numRAAN,num sma,numincl);
max_acc_Bre = zeros(numtimes,numRAAN,num sma,numincl);
max_acc_Inu = zeros(numtimes,numRAAN,num sma,numincl);
num_acc_Bre = zeros(numtimes,numRAAN,num sma,numincl);
num_acc_Inu = zeros(numtimes,numRAAN,num sma,numincl);

% time step in numerical and string format
time_step = 60;

% solar panel area in m^2 (14 cells with 26.62cm^2 each)
pan_area = 0.037268;

% vectors for panel normals in body frame, 0° and 45°, standard config
up_vec = [ 0 0 -1];
sq2h = sqrt(2)/2;
norm_vecs = zeros(8,3);
norm_vecs(1,:) = [ 1 0 0]; % +x
norm_vecs(2,:) = [-1 0 0]; % -x
norm_vecs(3,:) = [ 0 1 0]; % +y
norm_vecs(4,:) = [ 0 -1 0]; % -y
norm_vecs(5,:) = [ sq2h sq2h 0]; % +xy -> +x
norm_vecs(6,:) = [-sq2h -sq2h 0]; % -xy -> -x
norm_vecs(7,:) = [-sq2h sq2h 0]; % -x+y -> +y

```

```

norm_vecs(8,:) = [ sq2h -sq2h 0];    % +x-y  -> -y

% vectors for panel normals in body frame, 0° and 45°, deploy config 60°
norm_vecs_depl = zeros(8,3);
sq3h = sqrt(3)/2;
sq2v = sqrt(2)/4;
norm_vecs_depl(1,:) = [ 0.5  0 -sq3h];    % +x
norm_vecs_depl(2,:) = [-0.5  0 -sq3h];    % -x
norm_vecs_depl(3,:) = [ 0  0.5 -sq3h];    % +y
norm_vecs_depl(4,:) = [ 0 -0.5 -sq3h];    % -y
norm_vecs_depl(5,:) = [ sq2v  sq2v -sq3h]; % +xy  -> +x
norm_vecs_depl(6,:) = [-sq2v -sq2v -sq3h]; % -xy  -> -x
norm_vecs_depl(7,:) = [-sq2v  sq2v -sq3h]; % -x+y  -> +y
norm_vecs_depl(8,:) = [ sq2v -sq2v -sq3h]; % +x-y  -> -y

%-----
display('Loops over all cases...')
%-----
count = 0;
for ki = 1:numincl
for ka = 1:numsma
    % orbital period
    tU = 2*pi*sqrt(sma(ka)^3/3.986e14);
for kR = 1:numRAAN
for kt = 1:numtimes
    % define satellite properties and propagate
    Satellite.Propagator.StartTime = sat_start_time(kt);
    Satellite.Propagator.StopTime = sat_stop_time(kt);
    Satellite.Propagator.InitialState.Epoch = sat_start_time(kt);
    Satellite.Propagator.InitialState.Representation.AssignClassical ...
    ... ('eCoordinateSystemJ2000',sma(ka),0.0,incl(ki),0.0,RAAN_deg(kR),0.0);
    Satellite.Propagator.Propagate();
    %-----
    % get access times
    Access_Bre = Bremen.GetAccessToObject(Satellite);
    Access_Inu = Inuvik.GetAccessToObject(Satellite);
    Access_Bre.AccessTimePeriod = 'eObjectAccessTime';
    Access_Inu.AccessTimePeriod = 'eObjectAccessTime';
    Access_Bre.ComputeAccess;
    Access_Inu.ComputeAccess;
    DataProvider_Bre = Access_Bre.DataProviders.Item('Access Data');
    DataProvider_Inu = Access_Inu.DataProviders.Item('Access Data');
    Results_Bre = DataProvider_Bre.ExecElements(Satellite.Propagator.StartTime, ...
    ... Satellite.Propagator.StopTime, RptElems_acc);
    Results_Inu = DataProvider_Inu.ExecElements(Satellite.Propagator.StartTime, ...
    ... Satellite.Propagator.StopTime, RptElems_acc);
    Data_Bre = Results_Bre.DataSets;
    Data_Inu = Results_Inu.DataSets;

    % save data for Bremen
    if Data_Bre.RowCount > 0
        num_access_Bre = Data_Bre.RowCount;
        duration_Bre = zeros(num_access_Bre,1);
        dur_Bre = Data_Bre.GetDataSetByName('Duration').GetValues;
        for i = 1:num_access_Bre
            duration_Bre(i) = dur_Bre{i}(1);
        end
        acc_dur_Bre(kt,kR,ka,ki) = sum(duration_Bre);
        max_acc_Bre(kt,kR,ka,ki) = max(duration_Bre);
        num_acc_Bre(kt,kR,ka,ki) = num_access_Bre;
    else
        acc_dur_Bre(kt,kR,ka,ki) = 0;
        max_acc_Bre(kt,kR,ka,ki) = 0;
        num_acc_Bre(kt,kR,ka,ki) = 0;
    end
    % save data for Inuvik
    if Data_Inu.RowCount > 0
        num_access_Inu = Data_Inu.RowCount;
        duration_Inu = zeros(num_access_Inu,1);
        dur_Inu = Data_Inu.GetDataSetByName('Duration').GetValues;
        for i = 1:num_access_Inu
            duration_Inu(i) = dur_Inu{i}(1);
        end
        acc_dur_Inu(kt,kR,ka,ki) = sum(duration_Inu);
        max_acc_Inu(kt,kR,ka,ki) = max(duration_Inu);
        num_acc_Inu(kt,kR,ka,ki) = num_access_Inu;
    else
        acc_dur_Inu(kt,kR,ka,ki) = 0;
        max_acc_Inu(kt,kR,ka,ki) = 0;
        num_acc_Inu(kt,kR,ka,ki) = 0;
    end
    %-----
    % get lighting times
    Results = DataProvider.ExecElements(Satellite.Propagator.StartTime, ...
    ... Satellite.Propagator.StopTime, RptElems);
    Data = Results.DataSets;

    % save data in vectors
    num_light_per = Data.RowCount-2;    % leave out 1st and last value

```

```

start_time = zeros(num_light_per,1);
stop_time = zeros(num_light_per,1);
duration = zeros(num_light_per,1);
start_t = Data.GetDataSetByName('Start Time').GetValues;
stop_t = Data.GetDataSetByName('Stop Time').GetValues;
dur = Data.GetDataSetByName('Duration').GetValues;
for i = 1:num_light_per
    start_time(i) = start_t{i+1}(1);
    stop_time(i) = stop_t{i+1}(1);
    duration(i) = dur{i+1}(1);
end

% determine value and index of shortest and average light periods
[min_dur,min_ind] = min(duration);
[max_dur,~] = max(duration);

% average and max duration only of those daylight periods with t_light<t_U
sum_dur = 0;
amount_dur = 0;
max_dur_sm_tu = 0;
for i = 1:length(duration)
    if duration(i) < tU
        sum_dur = sum_dur+duration(i);
        amount_dur = amount_dur+1;
        % maximal duration smaller than tU and index
        if duration(i) > max_dur_sm_tu
            max_dur_sm_tu = duration(i);
            max_ind_sm_tu = i;
        end
    end
end
average_dur = sum_dur/amount_dur;
delta_dur = zeros(length(duration),1);
for i = 1:length(duration)
    delta_dur(i) = abs(duration(i)-average_dur);
end
[~,ave_ind] = min(delta_dur);
ave_dur = duration(ave_ind);

% prepare vectors for solar panel power
start_min_ave = [start_time(min_ind);start_time(ave_ind);start_time(max_ind_sm_tu)];
stop_min_ave = [stop_time(min_ind);stop_time(ave_ind);stop_time(max_ind_sm_tu)];
dur_min_ave = [min_dur,ave_dur,max_dur_sm_tu];

% save duration values
min_duration(kt,kR,ka,ki) = min_dur;
ave_duration(kt,kR,ka,ki) = ave_dur;
max_duration(kt,kR,ka,ki) = max_dur;
max_duration_sm_tu(kt,kR,ka,ki) = max_dur_sm_tu;

% prepare solar panel exposure calculations
% preallocate vector for energy output
energy = zeros(length(dur_min_ave),1);
energy_up = zeros(length(dur_min_ave),1);
energy_side = zeros(length(dur_min_ave),1);
energy_side_depl = zeros(length(dur_min_ave),1);
energy_depl = zeros(length(dur_min_ave),1);

% loop over minimum, maximum average daylight periods
for i=1:length(dur_min_ave)
    % calculate power results (time, vector to sun)
    Results_power = DataProvider.power.ExecElements ..
        ... (start_min_ave(i), stop_min_ave(i), time_step, ReportElements_power);
    Data_power = Results_power.DataSets;

    % calculate energy
    % save data in matrix vec
    time_vec = Data_power.GetDataSetByName('Time').GetValues;
    xMag_vec = Data_power.GetDataSetByName('x/Magnitude').GetValues;
    yMag_vec = Data_power.GetDataSetByName('y/Magnitude').GetValues;
    zMag_vec = Data_power.GetDataSetByName('z/Magnitude').GetValues;
    vec = zeros(Data_power.RowCount,length(Data_power.GetRow(1)));
    for j = 1:Data_power.RowCount
        vec(j,1) = time_vec{j}(1);
        vec(j,2) = xMag_vec{j}(1);
        vec(j,3) = yMag_vec{j}(1);
        vec(j,4) = zMag_vec{j}(1);
    end

    % save data in vectors
    % general
    time = zeros(Data_power.RowCount-1,1);
    power_total = zeros(Data_power.RowCount-1,1);
    power_side = zeros(Data_power.RowCount-1,1);
    power_total_depl = zeros(Data_power.RowCount-1,1);
    power_side_depl = zeros(Data_power.RowCount-1,1);
    sun_vector = zeros(Data_power.RowCount-1,3);

    % top panel

```

```

incid_angle_up = zeros(Data_power.RowCount-1,1);
angle_factor_up = zeros(Data_power.RowCount-1,1);
power_up       = zeros(Data_power.RowCount-1,1);
ang_sun_pan_up = zeros(Data_power.RowCount-1,1);

% side panels standard configuration
incid_angle = zeros(Data_power.RowCount-1,8);
angle_factor = zeros(Data_power.RowCount-1,8);
power       = zeros(Data_power.RowCount-1,8);
ang_sun_pan = zeros(Data_power.RowCount-1,8);

% side panels configuration with deployable panels
incid_angle_depl = zeros(Data_power.RowCount-1,8);
angle_factor_depl = zeros(Data_power.RowCount-1,8);
power_depl       = zeros(Data_power.RowCount-1,8);
ang_sun_pan_depl = zeros(Data_power.RowCount-1,8);

% loop over every time step
en_total = 0;
en_up    = 0;
en_side  = 0;
en_total_depl = 0;
en_side_depl = 0;

for j=1:(Data_power.RowCount-1)
    % general data
    time(j) = vec(j,1);
    sun_vector(j,:) = vec(j,2:4);

    % top panel
    ang_sun_pan_up(j) = acos(dot(sun_vector(j,:),up_vec));
    if ang_sun_pan_up(j) > pi/2
        incid_angle_up(j) = pi/2;
    else
        incid_angle_up(j) = ang_sun_pan_up(j);
    end
    if incid_angle_up(j) > (88.8*pi/180)
        angle_factor_up(j) = 0;
    elseif incid_angle_up(j) < (88.8*pi/180) && incid_angle_up(j) > (50*pi/180)
        angle_factor_up(j) = -0.369*(cos(incid_angle_up(j)))^3+ ...
        ... 0.637*(cos(incid_angle_up(j)))^2+0.750*cos(incid_angle_up(j))-0.015;
    else
        angle_factor_up(j) = cos(incid_angle_up(j));
    end
    power_up(j) = 1325.8*pan_area*angle_factor_up(j)*0.1977;

    % loop over side panels (4x0° spin offset, 4x45° spin offset)
    % standard configuration only for i=1 (minimum duration -> min
    % energy) and i=2 (average duration)
    if i==1 || i==2
        for pan=1:8
            ang_sun_pan(j,pan) = acos(dot(sun_vector(j,:),norm_vecs(pan,:)));
            if ang_sun_pan(j,pan) > pi/2
                incid_angle(j,pan) = pi/2;
            else
                incid_angle(j,pan) = ang_sun_pan(j,pan);
            end
            if incid_angle(j,pan) > (88.8*pi/180)
                angle_factor(j,pan) = 0;
            elseif incid_angle(j,pan) < (88.8*pi/180) && ...
                ... incid_angle(j,pan) > (50*pi/180)
                angle_factor(j,pan) = -0.369*(cos(incid_angle(j,pan)))^3+ ...
                ... 0.637*(cos(incid_angle(j,pan)))^2+ ...
                ... 0.750*cos(incid_angle(j,pan))-0.015;
            else
                angle_factor(j,pan) = cos(incid_angle(j,pan));
            end
            power(j,pan) = 1325.8*pan_area*angle_factor(j,pan)*0.1977;
        end

        % total power, side panels: average of 0° and 45° spin offset
        power_total(j) = power_up(j)+sum(power(j,:))/2;
        power_side(j) = sum(power(j,:))/2;
        en_up = en_up+power_up(j)*time_step/3600;
        en_side = en_side+power_side(j)*time_step/3600;
        en_total = en_total+power_total(j)*time_step/3600;
    end

    % loop over side panels (4x0° spin offset, 4x45° spin offset)
    % deploy configuration only for i=2 (average case) and i=3
    % (maximum duration smaller than tU -> min energy)
    if i==2 || i==3
        for pan=1:8
            ang_sun_pan_depl(j,pan) = acos(dot(sun_vector(j,:),norm_vecs_depl(pan,:)));
            if ang_sun_pan_depl(j,pan) > pi/2
                incid_angle_depl(j,pan) = pi/2;
            else
                incid_angle_depl(j,pan) = ang_sun_pan_depl(j,pan);
            end
        end
    end
end

```

```

        if incid_angle_depl(j,pan) > (88.8*pi/180)
            angle_factor_depl(j,pan) = 0;
        elseif incid_angle_depl(j,pan) < (88.8*pi/180) && ...
            ... incid_angle_depl(j,pan) > (50*pi/180)
            angle_factor_depl(j,pan) = -0.369*(cos(incid_angle_depl(j,pan)))^3+ ...
            ... 0.637*(cos(incid_angle_depl(j,pan)))^2+ ...
            ... 0.750*cos(incid_angle_depl(j,pan))-0.015;
        else
            angle_factor_depl(j,pan) = cos(incid_angle_depl(j,pan));
        end
        power_depl(j,pan) = 1325.8*pan_area*angle_factor_depl(j,pan)*0.1977;
    end

    % total power, side panels: average of 0° and 45° spin offset
    power_total_depl(j) = power_up(j)+sum(power_depl(j,:))/2;
    power_side_depl(j) = sum(power_depl(j,:))/2;
    en_side_depl = en_side_depl+power_side_depl(j)*time_step/3600;
    en_total_depl = en_total_depl+power_total_depl(j)*time_step/3600;
end

% save calculated energy for this daylight period in energy vector
energy(i) = en_total;
energy_up(i) = en_up;
energy_side(i) = en_side;
energy_depl(i) = en_total_depl;
energy_side_depl(i) = en_side_depl;
end

% save energy values in vectors
% for deploy configuration: minimum energy for maximum lighting time,
% that is shorter than tU, i=3 in the above for-loop
min_energy(kT,kR,ka,ki) = energy(1);
ave_energy(kT,kR,ka,ki) = energy(2);
min_energy_up(kT,kR,ka,ki) = energy_up(1);
ave_energy_up(kT,kR,ka,ki) = energy_up(2);
min_energy_side(kT,kR,ka,ki) = energy_side(1);
ave_energy_side(kT,kR,ka,ki) = energy_side(2);
min_energy_depl(kT,kR,ka,ki) = energy_depl(3);
ave_energy_depl(kT,kR,ka,ki) = energy_depl(2);
min_energy_side_depl(kT,kR,ka,ki) = energy_side_depl(3);
ave_energy_side_depl(kT,kR,ka,ki) = energy_side_depl(2);
% display progress
count = count+1;
progress = count*100/(number)
end
end
end
end

% save variables to disk
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\CLAVIS_General\min_energy_CLAVIS.mat', ...
    ... 'min_energy*');
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\CLAVIS_General\ave_energy_CLAVIS.mat', ...
    ... 'ave_energy*');
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\CLAVIS_General\duration_CLAVIS.mat', ...
    ... '*_duration*');
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\CLAVIS_General\inputs_CLAVIS.mat', ...
    ... 'RAAN_deg','sat_start_time','sma','incl');
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\CLAVIS_General\access_CLAVIS.mat', '*acc_*');

% close scenario
Scenario.Unload;
out=1;

end

```

## F.1.2 Skript zur Berechnung der erzeugten Energie als Funktion der Sonnenlichtdauer

### compute\_power\_as\_func\_of\_duration.m

```

function [out]=compute_power_as_func_of_duration()

%-----
display('Hook to running STK...')
%-----

uiApp = actxGetRunningServer('STK9.application');
root = uiApp.Personality2;

% define units
root.UnitPreferences.SetCurrentUnit('DateFormat','JDate');
root.UnitPreferences.SetCurrentUnit('DistanceUnit','m');
root.UnitPreferences.SetCurrentUnit('TimeUnit','sec');
root.UnitPreferences.SetCurrentUnit('AngleUnit','deg');

```

```

%-----
display('Create new objects and set their properties...')
%-----
% create new scenario, define scenario object
root.NewScenario('CLAVIS_Matlab_neu');
Scenario = root.GetObjectFromPath('*/*');
jd_start = jday(2012,01,01,00,00,00);
jd_stop = jday(2013,01,01,00,00,00);
Scenario.StopTime = jd_stop;
Scenario.StartTime = jd_start;

% use standard terrain data file
Scenario.TerrainCollection.Item(0).UseTerrain = 1;

% remove standard 2D and 3D windows
root.ExecuteCommand('Window3D * Remove WindowID 1');
root.ExecuteCommand('Window2D * Remove WindowID 1');

% maximize and hide main window
root.ExecuteCommand('Application / Maximize');
root.ExecuteCommand('Application / Hide');

% create Satellite CLAVIS
Satellite = Scenario.Children.New('eSatellite','CLAVIS');
Satellite.SetPropagatorType('ePropagatorJ2Perturbation');
Satellite.Propagator.StartTime = jd_start;
Satellite.Propagator.StopTime = jd_stop;

% set attitude and model file for satellite
Satellite.SetAttitudeType('eAttitudeStandard');
Satellite.Attitude.Basic.SetProfileType('eProfileSpinAboutNadir');
Satellite.Attitude.Basic.Profile.Rate = 0;
Satellite.Attitude.Basic.Profile.Offset = 0;

Satellite.VO.Model.ModelType = 'eModelFile';
Satellite.VO.Model.ModelData.FileName = 'STKData\VO\Models\Space\CLAVIS_Standard.mdl';

% only Earth as eclipse body
if Satellite.EclipseBodies.IsCentralBodyAssigned('Moon') == 1
    Satellite.EclipseBodies.UseCustomizedList = 1;
    Satellite.EclipseBodies.AssignCentralBody('Earth');
end

%-----
display('prepare outputs/report...')
%-----
% report content for lighting times
DataProvider = Satellite.DataProviders.Item('Lighting Times').Group.Item('Sunlight');
RptElems = {'Start Time'; 'Stop Time'; 'Duration'};

% report content for power calc, unit sun vector in body frame
DataProvider_power = Satellite.DataProviders.Item('Vectors(Body)').Group.Item('Sun');
ReportElements_power = {'Time'; 'x/Magnitude'; 'y/Magnitude'; 'z/Magnitude'};

% input altitude
sma = 6578140:50000:7028140;
numisma = length(sma);

% input inclination
incl = 0:15:90;
numincl = length(incl);

% max number of daylight periods
numlight = ceil((365.25*24*3600)/(2*pi*sqrt(6578140^3/3.986e14)))+1;

% number of runs
number = numisma*numincl;

% output vectors power
energy_stand_tot = zeros(numisma,numincl,numlight);
energy_top = zeros(numisma,numincl,numlight);
energy_depl_tot = zeros(numisma,numincl,numlight);
duration_light = zeros(numisma,numincl,numlight);

% time step in numerical and string format
time_step = 60;

% solar panel area in m^2 (14 cells with 26.62cm^2 each)
pan_area = 0.037268;

% vectors for panel normals in body frame, 0° and 45°, standard config
up_vec = [ 0 0 -1];
sq2h = sqrt(2)/2;
norm_vecs = zeros(8,3);
norm_vecs(1,:) = [ 1 0 0]; % +x
norm_vecs(2,:) = [-1 0 0]; % -x
norm_vecs(3,:) = [ 0 1 0]; % +y
norm_vecs(4,:) = [ 0 -1 0]; % -y

```



```

norm_vecs(5,:) = [ sq2h sq2h 0]; % +xy -> +x
norm_vecs(6,:) = [-sq2h -sq2h 0]; % -xy -> -x
norm_vecs(7,:) = [-sq2h sq2h 0]; % -x+y -> +y
norm_vecs(8,:) = [ sq2h -sq2h 0]; % +x-y -> -y

% vectors for panel normals in body frame, 0° and 45°, deploy config 60°
norm_vecs_depl = zeros(8,3);
sq3h = sqrt(3)/2;
sq2v = sqrt(2)/4;
norm_vecs_depl(1,:) = [ 0.5 0 -sq3h]; % +x
norm_vecs_depl(2,:) = [-0.5 0 -sq3h]; % -x
norm_vecs_depl(3,:) = [ 0 0.5 -sq3h]; % +y
norm_vecs_depl(4,:) = [ 0 -0.5 -sq3h]; % -y
norm_vecs_depl(5,:) = [ sq2v sq2v -sq3h]; % +xy -> +x
norm_vecs_depl(6,:) = [-sq2v -sq2v -sq3h]; % -xy -> -x
norm_vecs_depl(7,:) = [-sq2v sq2v -sq3h]; % -x+y -> +y
norm_vecs_depl(8,:) = [ sq2v -sq2v -sq3h]; % +x-y -> -y

%-----
display('Loops over all cases...')
%-----
count = 0;
for ki = 1:numincl
for ka = 1:numsma
    % define satellite properties and propagate
    Satellite.Propagator.InitialState.Representation.AssignClassical ...
    ... ('eCoordinateSystemJ2000',sma(ka),0.0,incl(ki),0.0,0.0,0.0);
    Satellite.Propagator.Propagate();
    %-----
    % get lighting times
    Results = DataProvider.ExecElements ...
    ... (Satellite.Propagator.StartTime, Satellite.Propagator.StopTime, RptElems);
    Data = Results.DataSets;

    % save data in vectors
    num_light_per = Data.RowCount-2; % leave out 1st and last value
    start_time = zeros(num_light_per,1);
    stop_time = zeros(num_light_per,1);
    duration = zeros(num_light_per,1);
    start_t = Data.GetDataSetByName('Start Time').GetValues;
    stop_t = Data.GetDataSetByName('Stop Time').GetValues;
    dur = Data.GetDataSetByName('Duration').GetValues;
    for i = 1:num_light_per
        start_time(i) = start_t{i+1}(1);
        stop_time(i) = stop_t{i+1}(1);
        duration(i) = dur{i+1}(1);
        % save duration values for output
        duration_light(ka,ki,i) = duration(i);
    end

    % loop over minimum, maximum average daylight periods
    for i=1:num_light_per
        % save report output
        Results_power = DataProvider_power.ExecElements ...
        ... (start_time(i), stop_time(i), time_step, ReportElements_power);
        Data_power = Results_power.DataSets;

        % calculate energy
        % save data in matrix vec
        time_vec = Data_power.GetDataSetByName('Time').GetValues;
        xMag_vec = Data_power.GetDataSetByName('x/Magnitude').GetValues;
        yMag_vec = Data_power.GetDataSetByName('y/Magnitude').GetValues;
        zMag_vec = Data_power.GetDataSetByName('z/Magnitude').GetValues;
        vec = zeros(Data_power.RowCount,length(Data_power.GetRow(1)));
        for j = 1:Data_power.RowCount
            vec(j,1) = time_vec{j}(1);
            vec(j,2) = xMag_vec{j}(1);
            vec(j,3) = yMag_vec{j}(1);
            vec(j,4) = zMag_vec{j}(1);
        end

        % save data in vectors
        % general
        time = zeros(Data_power.RowCount-1,1);
        power_total = zeros(Data_power.RowCount-1,1);
        power_total_depl = zeros(Data_power.RowCount-1,1);
        sun_vector = zeros(Data_power.RowCount-1,3);

        % top panel
        incid_angle_up = zeros(Data_power.RowCount-1,1);
        angle_factor_up = zeros(Data_power.RowCount-1,1);
        power_up = zeros(Data_power.RowCount-1,1);
        ang_sun_pan_up = zeros(Data_power.RowCount-1,1);

        % side panels standard configuration
        incid_angle = zeros(Data_power.RowCount-1,8);
        angle_factor = zeros(Data_power.RowCount-1,8);
        power = zeros(Data_power.RowCount-1,8);
        ang_sun_pan = zeros(Data_power.RowCount-1,8);

```

```

% side panels configuration with deployable panels
incid_angle_depl = zeros(Data_power.RowCount-1,8);
angle_factor_depl = zeros(Data_power.RowCount-1,8);
power_depl       = zeros(Data_power.RowCount-1,8);
ang_sun_pan_depl = zeros(Data_power.RowCount-1,8);

%loop over every time step
en_total = 0;
en_up    = 0;
en_total_depl = 0;

for j=1:(Data_power.RowCount-1)
    % general data
    time(j) = vec(j,1);
    sun_vector(j,:) = vec(j,2:4);

    % top panel
    ang_sun_pan_up(j) = acos(dot(sun_vector(j,:),up_vec));
    if ang_sun_pan_up(j) > pi/2
        incid_angle_up(j) = pi/2;
    else
        incid_angle_up(j) = ang_sun_pan_up(j);
    end
    if incid_angle_up(j) > (88.8*pi/180)
        angle_factor_up(j) = 0;
    elseif incid_angle_up(j) < (88.8*pi/180) && incid_angle_up(j) > (50*pi/180)
        angle_factor_up(j) = -0.369*(cos(incid_angle_up(j)))^3+ ...
            ... 0.637*(cos(incid_angle_up(j)))^2+0.750*cos(incid_angle_up(j))-0.015;
    else
        angle_factor_up(j) = cos(incid_angle_up(j));
    end
    power_up(j) = 1325.8*pan_area*angle_factor_up(j)*0.1977;

    % loop over side panels (4x0° spin offset, 4x45° spin offset)
    % standard configuration
    for pan=1:8
        ang_sun_pan(j,pan) = acos(dot(sun_vector(j,:),norm_vecs(pan,:)));
        if ang_sun_pan(j,pan) > pi/2
            incid_angle(j,pan) = pi/2;
        else
            incid_angle(j,pan) = ang_sun_pan(j,pan);
        end
        if incid_angle(j,pan) > (88.8*pi/180)
            angle_factor(j,pan) = 0;
        elseif incid_angle(j,pan) < (88.8*pi/180) && incid_angle(j,pan) > (50*pi/180)
            angle_factor(j,pan) = -0.369*(cos(incid_angle(j,pan)))^3+ ...
                ... 0.637*(cos(incid_angle(j,pan)))^2+0.750*cos(incid_angle(j,pan))-0.015;
        else
            angle_factor(j,pan) = cos(incid_angle(j,pan));
        end
        power(j,pan) = 1325.8*pan_area*angle_factor(j,pan)*0.1977;
    end

    % total power, side panels: average of 0° and 45° spin offset
    power_total(j) = power_up(j)+sum(power(j,:))/2;
    en_up          = en_up+power_up(j)*time_step/3600;
    en_total       = en_total+power_total(j)*time_step/3600;

    % loop over side panels (4x0° spin offset, 4x45° spin offset)
    % configuration with deployable panels
    for pan=1:8
        ang_sun_pan_depl(j,pan) = acos(dot(sun_vector(j,:),norm_vecs_depl(pan,:)));
        if ang_sun_pan_depl(j,pan) > pi/2
            incid_angle_depl(j,pan) = pi/2;
        else
            incid_angle_depl(j,pan) = ang_sun_pan_depl(j,pan);
        end
        if incid_angle_depl(j,pan) > (88.8*pi/180)
            angle_factor_depl(j,pan) = 0;
        elseif incid_angle_depl(j,pan) < (88.8*pi/180) && ...
            ... incid_angle_depl(j,pan) > (50*pi/180)
            angle_factor_depl(j,pan) = -0.369*(cos(incid_angle_depl(j,pan)))^3+ ...
                ... 0.637*(cos(incid_angle_depl(j,pan)))^2+ ...
                ... 0.750*cos(incid_angle_depl(j,pan))-0.015;
        else
            angle_factor_depl(j,pan) = cos(incid_angle_depl(j,pan));
        end
        power_depl(j,pan) = 1325.8*pan_area*angle_factor_depl(j,pan)*0.1977;
    end

    % total power, side panels: average of 0° and 45° spin offset
    power_total_depl(j) = power_up(j)+sum(power_depl(j,:))/2;
    en_total_depl       = en_total_depl+power_total_depl(j)*time_step/3600;

end

% save calculated energy for this daylight period in energy vector
energy_stand_tot(ka,ki,i) = en_total;
energy_top(ka,ki,i)       = en_up;

```

```

        energy_depl_tot(ka,ki,i) = en_total_depl;
    end
    % display progress
    count = count+1;
    progress = count*100/(number)
end
end

% save variables to disk
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\Energy_as_function_of_sunlight_duration\ ...
... energy_en_vs_dur.mat', 'energy_*', 'duration_light');
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\Energy_as_function_of_sunlight_duration\ ...
... inputs_en_vs_dur.mat', 'incl', 'sma');

% close scenario
Scenario.Unload;
out=1;
end

```

## F.2 Skripte zur Auswertung der Simulationsergebnisse

### F.2.1 Auswertung der Sonnenlichtzeiten und der erzeugten Energie als Funktion der Bahnparameter

Das hier gezeigte Skript zeigt die Auswertung der von der Standardkonfiguration erzeugten Energie als Funktion der Bahnparameter für Orbits mit Inklinationen zwischen  $0^\circ$  und  $90^\circ$ , inklusive der Bestimmung der Näherungsgleichungen. Die dabei verwendete Funktion *myfit\_linear* ist im Anschluss wiedergegeben. Die in den folgenden Skripten auftretenden Matrizen *fit\_matrix* werden als Eingabe für das online Tool [29] verwendet. Diese Skripte sind hier nur beispielhaft; es existieren ähnliche Skripte für die Konfigurationen mit ausklappbaren Paneelen sowie für SSOs. Alle verwendeten Skripte finden sich auf der CD zur Arbeit.

#### CLAVIS\_Standard\_auswertung.m

```

close all
clear all
clc

load 'G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\CLAVIS_General\min_energy_CLAVIS.mat'
load 'G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\CLAVIS_General\ave_energy_CLAVIS.mat'
load 'G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\CLAVIS_General\duration_CLAVIS.mat'
load 'G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\CLAVIS_General\inputs_CLAVIS.mat'

numRAAN = length(RAAN_deg);
numsma = length(sma);
numincl = length(incl);
numtimes = length(sat_start_time);

alt = sma./1000-6378.14;

% plot total average energy as fct of alt and incl
% Standard config
ave_energy_incl_sma = zeros(numsma,numincl);
ave_en_top = zeros(numsma,numincl);
ave_en_sides = zeros(numsma,numincl);
ave_en_tot_sum = zeros(numsma,numincl);
ave_en_top_sum = zeros(numsma,numincl);
ave_en_side_sum = zeros(numsma,numincl);
fit_matrix_top = zeros(numsma*numincl*numRAAN*numtimes,3);
fit_matrix_sides = zeros(numsma*numincl*numRAAN*numtimes,3);
count = 0;
for g = 1:numtimes
    for h = 1:numRAAN
        for i = 1:numsma
            for j = 1:numincl
                count = count+1;
                ave_energy_incl_sma(i,j) = ave_energy(g,h,i,j);
                ave_en_sides(i,j) = ave_energy_side(g,h,i,j);
                ave_en_top(i,j) = ave_energy_up(g,h,i,j);
                fit_matrix_top(count,1) = sma(i)/1000-6378.14;
                fit_matrix_top(count,2) = incl(j);
            end
        end
    end
end

```

```

        fit_matrix_top(count,3) = ave_energy_up(g,h,i,j);
        fit_matrix_sides(count,1) = sma(i)/1000-6378.14;
        fit_matrix_sides(count,2) = incl(j);
        fit_matrix_sides(count,3) = ave_energy_side(g,h,i,j);
    end
    end
    ave_en_tot_sum = ave_en_tot_sum+ave_energy_incl_sma;
    ave_en_top_sum = ave_en_top_sum+ave_en_top;
    ave_en_side_sum = ave_en_side_sum+ave_en_sides;
    mesh(alt,incl,ave_energy_incl_sma');
    hold on
    mesh(alt,incl,ave_en_sides')
    hold on
    mesh(alt,incl,ave_en_top')
    hold on
end
end
ylabel('inclination [deg]')
xlabel('altitude [km]')
zlabel('average generated energy (top, sides, total) [Wh]')
hold off

% average values for final plot
ave_en_tot = ave_en_tot_sum/(numtimes*numRAAN);
ave_en_top = ave_en_top_sum/(numtimes*numRAAN);
ave_en_sides = ave_en_side_sum/(numtimes*numRAAN);
figure
mesh(alt,incl,ave_en_tot');
hold on
mesh(alt,incl,ave_en_sides')
hold on
mesh(alt,incl,ave_en_top')
set(gca,'FontName','Palatino','FontSize',12)
colorbar
xlabel('Bahnhöhe [km]')
ylabel('Inklination [°]')
zlabel('erzeugte Energie [Wh]')
view(-53,14)
set(gca,'YTick',0:30:90,'XTick',200:150:650)
annotation('textbox',[0.4 0.1 0.1 0.1],'string','oberes Paneel', ...
    ... 'LineStyle','none','FontName','Palatino','FontSize',12)
annotation('textbox',[0.13 0.38 0.1 0.1],'string', ...
    ... 'seitliche Paneele','LineStyle','none','FontName','Palatino','FontSize',12)
annotation('textbox',[0.68 0.7 0.1 0.1],'string','gesamt', ...
    ... 'LineStyle','none','FontName','Palatino','FontSize',12)
grid on
hold off

% plot average values in 2D
figure
count = 0;
count_alt = 0;
p = zeros(3*4,1);
color_vec = ['k','b','r','g'];
for i = 1:numsma
    count_alt = count_alt+1;
    count = count+1;
    p(count) = plot(incl,ave_en_tot(i,:),'--','LineWidth',2,'Color',color_vec(count_alt));
    hold on
    count = count+1;
    p(count) = plot(incl,ave_en_sides(i,:),'--','LineWidth',2,'Color',color_vec(count_alt));
    hold on
    count = count+1;
    p(count) = plot(incl,ave_en_top(i,:),'-.','LineWidth',2,'Color',color_vec(count_alt));
    hold on
end
set(gca,'FontName','Palatino','FontSize',12)
legend([p(1) p(2) p(3) p(10) p(7) p(4) p(1)],'gesamt','seitliche Paneele', ...
    ... 'oberes Paneel','h=650km','h=500km','h=350km','h=200km','Location','EastOutside')
xlabel('Inklination [°]')
xlim([0 90])
ylabel('erzeugte Energie [Wh]')
grid on
hold off

% reduce size of fit matrices by 1/8
new_length = 8000;
new_length = floor(numsma*numincl*numRAAN*numtimes/8);
fit_matrix_top_new = zeros(new_length,3);
fit_matrix_sides_new = zeros(new_length,3);
count = 0;
for i = 1:numsma*numincl*numRAAN*numtimes
    if mod(i,8) == 0
        count = count+1;
        fit_matrix_top_new(count,:) = fit_matrix_top(i,:);
        fit_matrix_sides_new(count,:) = fit_matrix_sides(i,:);
    end
end
end
fit_matrix_top_new = sortrows(fit_matrix_top_new);

```

```

fit_matrix_sides_new = sortrows(fit_matrix_sides_new);

% top
figure
plot3(fit_matrix_top_new(:,1),fit_matrix_top_new(:,2),fit_matrix_top_new(:,3),'.')
xlabel('altitude [km]')
ylabel('inclination [deg]')
zlabel('average generated energy [Wh]')
grid on
hold on
% fitted function
incl_fit = 0:10:90;
alt_fit = 200:50:650;
l_incl = length(incl_fit);
l_alt = length(alt_fit);
en_top_fit = zeros(l_alt,l_incl);

start_values = [4.3418660064521770E+00
7.6684807742465316E-04
-2.6336670069423630E-02
-9.4458074482722503E-08
2.1372100358955692E-05
1.2419596519754286E-05];

options = optimset('Display','iter','MaxFunEvals',20000, ...
... 'MaxIter',10000,'TolX',1e-8,'TolFun',1e-8);
Estimates = fminsearch(@myfit_linear,start_values,options, ...
... fit_matrix_top(:,1),fit_matrix_top(:,2),fit_matrix_top(:,3));
a = Estimates(1)
b = Estimates(2)
c = Estimates(3)

% plot fit
for i = 1:l_alt
    for j = 1:l_incl
        x = alt_fit(i);
        y = incl_fit(j);
        en_top_fit(i,j) = a+b*x+c*y;
    end
end
mesh(alt_fit,incl_fit,en_top_fit')
hold off
% error statistics
error_matrix = zeros(numsma*numincl*numtimes*numRAAN,3);
count = 0;
for gg = 1:numtimes
    for hh = 1:numRAAN
        for ii = 1:numsma
            for jj = 1:numincl
                count = count+1;
                x = alt(ii);
                y = incl(jj);
                error_matrix(count,1) = x;
                error_matrix(count,2) = y;
                error_matrix(count,3) = (a+b*x+c*y-ave_energy_up(gg,hh,ii,jj))*100/ ...
                ... ave_energy_up(gg,hh,ii,jj);
            end
        end
    end
end
mean_error_top = (mean(abs(error_matrix(:,3))))
max_error_top = (max(abs(error_matrix(:,3))))

% sides
figure
plot3(fit_matrix_sides(:,1),fit_matrix_sides(:,2),fit_matrix_sides(:,3),'.')
xlabel('altitude [km]')
ylabel('inclination [deg]')
zlabel('average generated energy [Wh]')
grid on
hold on
% fitted function
incl_fit = 0:10:90;
alt_fit = 200:50:650;
l_incl = length(incl_fit);
l_alt = length(alt_fit);
en_side_fit = zeros(l_alt,l_incl);

start_values = [6.1813763920532123E+00
5.8976544597588009E-03
4.3968511657499362E-02];

options = optimset('Display','iter','MaxFunEvals',20000, ...
... 'MaxIter',10000,'TolX',1e-8,'TolFun',1e-8);
Estimates = fminsearch(@myfit_linear,start_values,options, ...
... fit_matrix_sides(:,1),fit_matrix_sides(:,2),fit_matrix_sides(:,3));
a = Estimates(1)
b = Estimates(2)
c = Estimates(3)

```

```

% plot fit
for i = 1:l_alt
    for j = 1:l_incl
        x = alt_fit(i);
        y = incl_fit(j);
        en_side_fit(i,j) = a+b*x+c*y;
    end
end
mesh(alt_fit,incl_fit,en_side_fit')
hold off
% error statistics
error_matrix = zeros(numsma*numincl*numtimes*numRAAN,3);
count = 0;
for gg = 1:numtimes
    for hh = 1:numRAAN
        for ii = 1:numsma
            for jj = 1:numincl
                count = count+1;
                x = alt(ii);
                y = incl(jj);
                error_matrix(count,1) = x;
                error_matrix(count,2) = y;
                error_matrix(count,3) = (a+b*x+c*y-ave_energy_side(gg,hh,ii,jj))*100/ ...
                ... ave_energy_side(gg,hh,ii,jj);
            end
        end
    end
end
mean_error = (mean(abs(error_matrix(:,3))))
max_error = (max(abs(error_matrix(:,3))))

% plot total energy again with fitted functions
figure
mesh(alt_fit,incl_fit,en_side_fit')
xlabel('altitude [km]')
ylabel('inclination [deg]')
zlabel('average generated energy [Wh]')
zlim([0 20])
hold on
mesh(alt_fit,incl_fit,en_top_fit')
hold on
mesh(alt_fit,incl_fit,(en_top_fit+en_side_fit'))
hold off

% compute reduction caused by missing panels
red_top_ave = 100-(en_side_fit.*100./(en_top_fit+en_side_fit));
red_side_ave = 100-((en_top_fit+0.75.*en_side_fit).*100./(en_top_fit+en_side_fit));

% -----
% minimum energy
% plot total average energy as fct of alt and incl
% Standard config
figure
min_energy_incl_sma = zeros(numsma,numincl);
min_en_top = zeros(numsma,numincl);
min_en_sides = zeros(numsma,numincl);
min_en_tot_sum = zeros(numsma,numincl);
min_en_top_sum = zeros(numsma,numincl);
min_en_side_sum = zeros(numsma,numincl);
fit_matrix_top = zeros(numsma*numincl*numRAAN*numtimes,3);
fit_matrix_sides = zeros(numsma*numincl*numRAAN*numtimes,3);
count = 0;
for g = 1:numtimes
    for h = 1:numRAAN
        for i = 1:numsma
            for j = 1:numincl
                count = count+1;
                min_energy_incl_sma(i,j) = min_energy(g,h,i,j);
                min_en_sides(i,j) = min_energy_side(g,h,i,j);
                min_en_top(i,j) = min_energy_up(g,h,i,j);
                fit_matrix_top(count,1) = sma(i)/1000-6378.14;
                fit_matrix_top(count,2) = incl(j);
                fit_matrix_top(count,3) = min_energy_up(g,h,i,j);
                fit_matrix_sides(count,1) = sma(i)/1000-6378.14;
                fit_matrix_sides(count,2) = incl(j);
                fit_matrix_sides(count,3) = min_energy_side(g,h,i,j);
            end
        end
    end
    min_en_tot_sum = min_en_tot_sum+min_energy_incl_sma;
    min_en_top_sum = min_en_top_sum+min_en_top;
    min_en_side_sum = min_en_side_sum+min_en_sides;
    mesh(alt,incl,min_energy_incl_sma');
    hold on
    mesh(alt,incl,min_en_sides')
    hold on
    mesh(alt,incl,min_en_top')
    hold on
end
end

```

```

end
ylabel('inclination [deg]')
xlabel('altitude [km]')
zlabel('minimum generated energy (top, sides, total) [Wh]')
hold off

% minimum values for final plot
min_en_tot = min_en_tot_sum./(numtimes*numRAAN);
min_en_top = min_en_top_sum./(numtimes*numRAAN);
min_en_sides = min_en_side_sum./(numtimes*numRAAN);
figure
mesh(alt,incl,min_en_tot');
hold on
mesh(alt,incl,min_en_sides')
hold on
mesh(alt,incl,min_en_top')
set(gca,'FontName','Palatino','FontSize',12)
colorbar
xlabel('Bahnhöhe [km]')
ylabel('Inklination [°]')
zlabel('erzeugte Energie [Wh]')
view(-53,14)
set(gca,'YTick',0:30:90,'XTick',200:150:650)
annotation('textbox',[0.61 0.23 0.1 0.1],'string','oberes Paneel', ...
... 'LineStyle','none','FontName','Palatino','FontSize',12)
annotation('textbox',[0.58 0.47 0.1 0.1],'string', ...
... 'seitliche Paneele','LineStyle','none','FontName','Palatino','FontSize',12)
annotation('textbox',[0.69 0.74 0.1 0.1],'string','gesamt', ...
... 'LineStyle','none','FontName','Palatino','FontSize',12)
grid on
hold off

% plot average values in 2D
figure
count = 0;
count_alt = 0;
p = zeros(3*4,1);
color_vec = ['k','b','r','g'];
for i = 1:3:numsma
    count_alt = count_alt+1;
    count = count+1;
    p(count) = plot(incl,min_en_tot(i,:),'--','LineWidth',2,'Color',color_vec(count_alt));
    hold on
    count = count+1;
    p(count) = plot(incl,min_en_sides(i,:),'--','LineWidth',2,'Color',color_vec(count_alt));
    hold on
    count = count+1;
    p(count) = plot(incl,min_en_top(i,:),'-.','LineWidth',2,'Color',color_vec(count_alt));
    hold on
end
set(gca,'FontName','Palatino','FontSize',12)
legend([p(1) p(2) p(3) p(4) p(5) p(6) p(7) p(8)], 'gesamt', 'seitliche Paneele', ...
... 'oberes Paneel', 'h=650km', 'h=500km', 'h=350km', 'h=200km', 'Location', 'EastOutside')
xlabel('Inklination [°]')
xlim([0 90])
ylabel('erzeugte Energie [Wh]')
grid on
hold off

% top
figure
plot3(fit_matrix_top(:,1),fit_matrix_top(:,2),fit_matrix_top(:,3),'.')
xlabel('altitude [km]')
ylabel('inclination [deg]')
zlabel('minimum generated energy [Wh]')
grid on
hold on
% fitted function
incl_fit = 0:10:90;
alt_fit = 200:50:650;
l_incl = length(incl_fit);
l_alt = length(alt_fit);
en_top_fit = zeros(l_alt,l_incl);

start_values = [4.3418660064521770E+00
7.6684807742465316E-04
-2.6336670069423630E-02];

options = optimset('Display','iter','MaxFunEvals',20000, ...
... 'MaxIter',10000,'TolX',1e-8,'TolFun',1e-8);
Estimates = fminsearch(@myfit_linear,start_values,options, ...
... fit_matrix_top(:,1),fit_matrix_top(:,2),fit_matrix_top(:,3));
a = Estimates(1)
b = Estimates(2)
c = Estimates(3)

% plot fit
for i = 1:l_alt
    for j = 1:l_incl

```

```

        x = alt_fit(i);
        y = incl_fit(j);
        en_top_fit(i,j) = a+b*x+c*y;
    end
end
mesh(alt_fit,incl_fit,en_top_fit')
hold off
% error statistics
error_matrix = zeros(numsma*numincl*numtimes*numRAAN,3);
count = 0;
for gg = 1:numtimes
    for hh = 1:numRAAN
        for ii = 1:numsma
            for jj = 1:numincl
                count = count+1;
                x = alt(ii);
                y = incl(jj);
                error_matrix(count,1) = x;
                error_matrix(count,2) = y;
                error_matrix(count,3) = (a+b*x+c*y-min_energy_up(gg,hh,ii,jj))*100/ ...
                    ... min_energy_up(gg,hh,ii,jj);
            end
        end
    end
end
mean_error_top = (mean(abs(error_matrix(:,3))))
max_error_top = (max(abs(error_matrix(:,3))))

% sides
figure
plot3(fit_matrix_sides(:,1),fit_matrix_sides(:,2),fit_matrix_sides(:,3),'.')
xlabel('altitude [km]')
ylabel('inclination [deg]')
zlabel('minimum generated energy [Wh]')
grid on
hold on
% fitted function
incl_fit = 0:10:90;
alt_fit = 200:50:650;
l_incl = length(incl_fit);
l_alt = length(alt_fit);
en_side_fit = zeros(l_alt,l_incl);

start_values = [6.1813763920532123E+00
    5.8976544597588009E-03
    4.3968511657499362E-02];

options = optimset('Display','iter','MaxFunEvals',20000, ...
    ... 'MaxIter',10000,'TolX',1e-8,'TolFun',1e-8);
Estimates = fminsearch(@myfit_linear,start_values,options, ...
    ... fit_matrix_sides(:,1),fit_matrix_sides(:,2),fit_matrix_sides(:,3));
a = Estimates(1)
b = Estimates(2)
c = Estimates(3)

% plot fit
for i = 1:l_alt
    for j = 1:l_incl
        x = alt_fit(i);
        y = incl_fit(j);
        en_side_fit(i,j) = a+b*x+c*y;
    end
end
mesh(alt_fit,incl_fit,en_side_fit')
hold off
% error statistics
error_matrix = zeros(numsma*numincl*numtimes*numRAAN,3);
count = 0;
for gg = 1:numtimes
    for hh = 1:numRAAN
        for ii = 1:numsma
            for jj = 1:numincl
                count = count+1;
                x = alt(ii);
                y = incl(jj);
                error_matrix(count,1) = x;
                error_matrix(count,2) = y;
                error_matrix(count,3) = (a+b*x+c*y-min_energy_side(gg,hh,ii,jj))*100/ ...
                    ... min_energy_side(gg,hh,ii,jj);
            end
        end
    end
end
mean_error_sides = (mean(abs(error_matrix(:,3))))
max_error_sides = (max(abs(error_matrix(:,3))))

% plot total energy again with fitted functions
figure
mesh(alt_fit,incl_fit,en_side_fit')

```



```

xlabel('altitude [km]')
ylabel('inclination [deg]')
zlabel('minimum generated energy [Wh]')
zlim([4 14])
hold on
mesh(alt_fit,incl_fit,en_top_fit')
hold on
mesh(alt_fit,incl_fit,(en_top_fit+en_side_fit'))
hold off

% compute reduction caused by missing panels
red_top_min = 100-(en_side_fit.*100./(en_top_fit+en_side_fit));
red_side_min = 100-((en_top_fit+0.75.*en_side_fit).*100./(en_top_fit+en_side_fit));

```

### Funktion *myfit\_linear.m*

```

function sse = myfit_linear(params,x,y,Actual_Output)

a = params(1);
b = params(2);
c = params(3);

% compute fitted curve
Fitted_Curve = a+b.*x+c.*y;

% compute error
Error_Vector = Fitted_Curve - Actual_Output;

% compute sum of squares error
sse = sum(Error_Vector.^2);

end

```

## F.2.2 Auswertung der erzeugten Energie als Funktion der Sonnenlichtdauer

Das hier wiedergegebene Skript zeigt die Auswertung der von der Standardkonfiguration erzeugten Energie als Funktion der Sonnenlichtdauer, ähnliche Skripte existieren für die Konfigurationen mit ausklappbaren Paneelen. Die hier verwendete Funktion *myfit\_rationalS* ähnelt der Funktion *myfit\_linear* und findet sich auf der CD zur Arbeit.

### CLAVIS\_Standard\_energy\_as\_func\_of\_duration.m

```

clear all
close all
clc

load('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\ ...
... Energy_as_function_of_sunlight_duration\energy_en_vs_dur.mat');
load('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\ ...
... Energy_as_function_of_sunlight_duration\inputs_en_vs_dur.mat');

num sma = length(sma);
num incl = length(incl);
num light = 5972;

% side panels
dur_en_matrix = zeros(numlight*numincl*numsma,3);
count = 0;
for i = 1:numsma
    for j = 1:numincl
        for k = 1:numlight
            count = count+1;
            dur_en_matrix(count,1) = duration_light(i,j,k)/60;
            dur_en_matrix(count,2) = sma(i)/1000-6378.14;
            dur_en_matrix(count,3) = energy_stand_tot(i,j,k)-energy_top(i,j,k);
        end
    end
end

% maximum orbit period
tmax = 2*pi*sqrt(7028140^3/3.986e14)/60;

% reduce data to valid interval, duration > 0 & duration < tmax
first_index = 0;
last_index = numlight;

```

```

dur_en_matrix = sortrows(dur_en_matrix);
for i = 2:numlight*numincl*numsma
    if dur_en_matrix(i-1,1) == 0 && dur_en_matrix(i,1) ~= 0
        first_index = i;
    elseif dur_en_matrix(i-1,1) < tmax && dur_en_matrix(i,1) > tmax
        last_index = i;
    end
end
% put reduced data in smaller matrix
new_length = last_index-first_index;
matrix_new = zeros(new_length,3);
for i = 1:new_length
    matrix_new(i,1) = dur_en_matrix(first_index+i-1,1);
    matrix_new(i,2) = dur_en_matrix(first_index+i-1,2);
    matrix_new(i,3) = dur_en_matrix(first_index+i-1,3);
end

% reduce matrix to only ca. 5000 rows, filter out many values at the
% beginning!
new_new_length = floor(new_length/75);
matrix_new_new = zeros(new_new_length,3);
count = 0;
for i = 1:(new_length-4000)
    if mod(i,100) == 0
        count = count+1;
        matrix_new_new(count,:) = matrix_new(i,:);
    end
end
for i = (new_length-3999):new_length
    count = count+1;
    matrix_new_new(count,:) = matrix_new(i,:);
end

% plot data in 2D
matrix_sides_plot = sortrows(matrix_new,2);
split_ind = zeros(10,1);
for i = 1:length(matrix_sides_plot)
    if matrix_sides_plot(i,2) == 200 && matrix_sides_plot(i+1,2) == 250
        split_ind(1) = i;
    elseif matrix_sides_plot(i,2) == 250 && matrix_sides_plot(i+1,2) == 300
        split_ind(2) = i;
    elseif matrix_sides_plot(i,2) == 300 && matrix_sides_plot(i+1,2) == 350
        split_ind(3) = i;
    elseif matrix_sides_plot(i,2) == 350 && matrix_sides_plot(i+1,2) == 400
        split_ind(4) = i;
    elseif matrix_sides_plot(i,2) == 400 && matrix_sides_plot(i+1,2) == 450
        split_ind(5) = i;
    elseif matrix_sides_plot(i,2) == 450 && matrix_sides_plot(i+1,2) == 500
        split_ind(6) = i;
    elseif matrix_sides_plot(i,2) == 500 && matrix_sides_plot(i+1,2) == 550
        split_ind(7) = i;
    elseif matrix_sides_plot(i,2) == 550 && matrix_sides_plot(i+1,2) == 600
        split_ind(8) = i;
    elseif matrix_sides_plot(i,2) == 600 && matrix_sides_plot(i+1,2) == 650
        split_ind(9) = i;
    end
end
split_ind(10) = length(matrix_sides_plot);
figure
count = 0;
count_alt = 0;
color_vec = ['k','b','r','g'];
for i = 1:3:numsma
    count_alt = count_alt+1;
    if i == 1
        count = count+1;
        plot(matrix_sides_plot(1:split_ind(1),1),matrix_sides_plot(1:split_ind(1),3), ...
            ... '.', 'Color', color_vec(count_alt));
        hold on
    else
        count = count+1;
        plot(matrix_sides_plot(split_ind(i-1):split_ind(i),1), ...
            ... matrix_sides_plot(split_ind(i-1):split_ind(i),3), '.', 'Color', color_vec(count_alt));
        hold on
    end
end
set(gca, 'FontName', 'Palatino', 'FontSize', 12)
legend('h=200km', 'h=350km', 'h=500km', 'h=650km', 'Location', 'SouthEast')
xlabel('Dauer [min]')
ylabel('erzeugte Energie [Wh]')
grid on
hold off

% fit data to curve
start_values = [1.8800150915350995E+00
-3.7926871383329952E-02
3.1307216674058538E-04
-1.1479415791621151E-06

```

```

-2.5850951692844620E-01
-7.3504142059831190E-02
1.9465464594717895E-02];
options = optimset('Display','iter','MaxFunEvals',20000, ...
... 'MaxIter',10000,'TolX',1e-8,'TolFun',1e-8);
Estimates = fminsearch(@myfit_rationalS,start_values,options, ...
... matrix_new(:,1),matrix_new(:,2),matrix_new(:,3));
a = Estimates(1)
b = Estimates(2)
c = Estimates(3)
d = Estimates(4)
f = Estimates(5)
g = Estimates(6)
h = Estimates(7)

% plot
figure
plot3(matrix_new_new(:,1),matrix_new_new(:,2),matrix_new_new(:,3),'.k')
set(gca,'FontName','Palatino','FontSize',12)
colorbar
xlabel('Dauer [min]')
ylabel('Bahnhöhe [km]')
zlabel('erzeugte Energie [Wh]')
xlim([51 98])
ylim([200 700])
zlim([5 20])
grid on
hold on
dur_fit = 51:1:98;
alt_fit = 200:10:650;
l_dur = length(dur_fit);
l_alt = length(alt_fit);
en_fit = zeros(l_dur,l_alt);
for i = 1:l_dur
    for j = 1:l_alt
        x = dur_fit(i);
        y = alt_fit(j);
        en_fit(i,j) = (a+b*x+c*y+d*x*y)/(1+f*log(x)+g*log(y)+h*log(x)*log(y));
        if en_fit(i,j) > 20 || en_fit(i,j) < 5
            en_fit(i,j) = 0/0;
        end
    end
end
mesh(dur_fit,alt_fit,en_fit')
set(gca,'Clim',[6 19])
hold off

% error
error = zeros(new_length,1);
for i = 1:new_length
    x = matrix_new(i,1);
    y = matrix_new(i,2);
    error(i) = ((a+b*x+c*y+d*x*y)/(1+f*log(x)+g*log(y)+h*log(x)*log(y))- ...
... matrix_new(i,3))*100/matrix_new(i,3);
end

% display error statistics
max_error_3D = max(abs(error))
mean_error_3D = mean(abs(error))

%-----
% top

dur_en_matrix_top = zeros(numlight*numincl*numsma,3);
count = 0;
for i = 1:numsma
    for j = 1:numincl
        for k = 1:numlight
            count = count+1;
            dur_en_matrix_top(count,1) = duration_light(i,j,k)/60;
            dur_en_matrix_top(count,2) = sma(i)/1000-6378.14;
            dur_en_matrix_top(count,3) = energy_top(i,j,k);
        end
    end
end

% maximum orbit period
tmax = 2*pi*sqrt(7028140^3/3.986e14)/60;

% reduce data to valid interval, duration > 0 & duration < tmax
first_index = 0;
last_index = numlight;

dur_en_matrix_top = sortrows(dur_en_matrix_top);
for i = 2:numlight*numincl*numsma
    if dur_en_matrix_top(i-1,1) == 0 && dur_en_matrix_top(i,1) ~= 0
        first_index = i;
    elseif dur_en_matrix_top(i-1,1) < tmax && dur_en_matrix_top(i,1) > tmax
        last_index = i;
    end
end

```

```

end
end
% put reduced data in smaller matrix
new_length = last_index-first_index;
matrix_new_top = zeros(new_length,3);
for i = 1:new_length
    matrix_new_top(i,1) = dur_en_matrix_top(first_index+i-1,1);
    matrix_new_top(i,2) = dur_en_matrix_top(first_index+i-1,2);
    matrix_new_top(i,3) = dur_en_matrix_top(first_index+i-1,3);
end

% plot data in 2D
matrix_sides_plot = sortrows(matrix_new_top,2);
split_ind = zeros(10,1);
for i = 1:length(matrix_sides_plot)
    if matrix_sides_plot(i,2) == 200 && matrix_sides_plot(i+1,2) == 250
        split_ind(1) = i;
    elseif matrix_sides_plot(i,2) == 250 && matrix_sides_plot(i+1,2) == 300
        split_ind(2) = i;
    elseif matrix_sides_plot(i,2) == 300 && matrix_sides_plot(i+1,2) == 350
        split_ind(3) = i;
    elseif matrix_sides_plot(i,2) == 350 && matrix_sides_plot(i+1,2) == 400
        split_ind(4) = i;
    elseif matrix_sides_plot(i,2) == 400 && matrix_sides_plot(i+1,2) == 450
        split_ind(5) = i;
    elseif matrix_sides_plot(i,2) == 450 && matrix_sides_plot(i+1,2) == 500
        split_ind(6) = i;
    elseif matrix_sides_plot(i,2) == 500 && matrix_sides_plot(i+1,2) == 550
        split_ind(7) = i;
    elseif matrix_sides_plot(i,2) == 550 && matrix_sides_plot(i+1,2) == 600
        split_ind(8) = i;
    elseif matrix_sides_plot(i,2) == 600 && matrix_sides_plot(i+1,2) == 650
        split_ind(9) = i;
    end
end
split_ind(10) = length(matrix_sides_plot);
figure
count = 0;
count_alt = 0;
color_vec = ['k','b','r','g'];
for i = 1:3:numsma
    count_alt = count_alt+1;
    if i == 1
        count = count+1;
        plot(matrix_sides_plot(1:split_ind(1),1),matrix_sides_plot(1:split_ind(1),3), ...
            ... '.', 'Color',color_vec(count_alt));
        hold on
    else
        count = count+1;
        plot(matrix_sides_plot(split_ind(i-1):split_ind(i),1), ...
            ... matrix_sides_plot(split_ind(i-1):split_ind(i),3), '.', 'Color',color_vec(count_alt));
        hold on
    end
end
set(gca,'FontName','Palatino','FontSize',12)
legend('h=200km','h=350km','h=500km','h=650km','Location','NorthEast')
xlabel('Dauer [min]')
ylabel('erzeugte Energie [Wh]')
grid on
hold off

% reduce matrix to only ca. 5000 rows, filter out many values at the
% beginning!
new_new_length = floor(new_length/75);
matrix_new_new_top = zeros(new_new_length,3);
count = 0;
for i = 1:(new_length-4000)
    if mod(i,100) == 0
        count = count+1;
        matrix_new_new_top(count,:) = matrix_new_top(i,:);
    end
end
for i = (new_length-3999):new_length
    count = count+1;
    matrix_new_new_top(count,:) = matrix_new_top(i,:);
end

% fit data to curve
start_values = [-9.4786780141961985E-02
    3.5309596460562069E-04
    1.0738949609713693E-05
    -9.6470825124173835E-07
    -2.7974131362124832E-01
    -9.1746834982347439E-02
    2.7232130219790929E-02];

options = optimset('Display','iter','MaxFunEvals',20000, ...
    ... 'MaxIter',10000,'TolX',1e-8,'TolFun',1e-8);
Estimates = fminsearch(@myfit_rationalS,start_values,options, ...

```

```

... matrix_new_new_top(:,1),matrix_new_new_top(:,2),matrix_new_new_top(:,3));
a = Estimates(1)
b = Estimates(2)
c = Estimates(3)
d = Estimates(4)
f = Estimates(5)
g = Estimates(6)
h = Estimates(7)

% plot
figure
plot3(matrix_new_new_top(:,1),matrix_new_new_top(:,2),matrix_new_new_top(:,3),'.k')
set(gca,'FontName','Palatino','FontSize',12)
set(gca,'YTick',200:100:600)
colorbar
xlabel('Dauer [min]')
ylabel('Bahnhöhe [km]')
zlabel('erzeugte Energie [Wh]')
xlim([51 98])
ylim([200 700])
zlim([0.8 5])
grid on
hold on
dur_fit = 51:1:98;
alt_fit = 200:10:650;
l_dur = length(dur_fit);
l_alt = length(alt_fit);
en_fit = zeros(l_dur,l_alt);
for i = 1:l_dur
    for j = 1:l_alt
        x = dur_fit(i);
        y = alt_fit(j);
        en_fit(i,j) = (a+b*x+c*y+d*x*y)/(1+f*log(x)+g*log(y)+h*log(x)*log(y));
        if en_fit(i,j) > 5 || en_fit(i,j) < 0
            en_fit(i,j) = 0/0;
        end
    end
end
mesh(dur_fit,alt_fit,en_fit)
ax_handle = gca;
set(ax_handle,'Clim',[0.8 5])
hold off

% error
error_top = zeros(new_length,1);
for i = 1:new_length
    x = matrix_new_top(i,1);
    y = matrix_new_top(i,2);
    error_top(i) = ((a+b*x+c*y+d*x*y)/(1+f*log(x)+g*log(y)+h*log(x)*log(y))- ...
        ... matrix_new_top(i,3))*100/matrix_new_top(i,3);
end

% display error statistics
max_error_3D_top = max(abs(error_top))
mean_error_3D_top = mean(abs(error_top))

%-----
% total 2D plot

dur_en_matrix_tot = zeros(numlight*numincl*numsma,3);
count = 0;
for i = 1:numsma
    for j = 1:numincl
        for k = 1:numlight
            count = count+1;
            dur_en_matrix_tot(count,1) = duration_light(i,j,k)/60;
            dur_en_matrix_tot(count,2) = sma(i)/1000-6378.14;
            dur_en_matrix_tot(count,3) = energy_stand_tot(i,j,k);
        end
    end
end

% maximum orbit period
tmax = 2*pi*sqrt(7028140^3/3.986e14)/60;

% reduce data to valid interval, duration > 0 & duration < tmax
first_index = 0;
last_index = numlight;

dur_en_matrix_tot = sortrows(dur_en_matrix_tot);
for i = 2:numlight*numincl*numsma
    if dur_en_matrix_tot(i-1,1) == 0 && dur_en_matrix_tot(i,1) ~= 0
        first_index = i;
    elseif dur_en_matrix_tot(i-1,1) < tmax && dur_en_matrix_tot(i,1) > tmax
        last_index = i;
    end
end
% put reduced data in smaller matrix
new_length = last_index-first_index;

```

```

matrix_new_tot = zeros(new_length,3);
for i = 1:new_length
    matrix_new_tot(i,1) = dur_en_matrix_tot(first_index+i-1,1);
    matrix_new_tot(i,2) = dur_en_matrix_tot(first_index+i-1,2);
    matrix_new_tot(i,3) = dur_en_matrix_tot(first_index+i-1,3);
end

matrix_new_tot = sortrows(matrix_new_tot,3);
% reduce size of matrix to limit size of plot *.eps-file,
% filter out many values at the beginning
count = 0;
for i = 1:(new_length-100000)
    if mod(i,150) == 0
        count = count+1;
        matrix_new_new_tot(count,:) = matrix_new_tot(i,:);
    end
end
for i = (new_length-99999):(new_length-3000)
    if mod(i,8) == 0
        count = count+1;
        matrix_new_new_tot(count,:) = matrix_new_tot(i,:);
    end
end
for i = (new_length-2999):new_length
    if mod(i,5) == 0
        count = count+1;
        matrix_new_new_tot(count,:) = matrix_new_tot(i,:);
    end
end
new_new_length = length(matrix_new_new_tot);
% sort by altitude
matrix_new_new_tot = sortrows(matrix_new_new_tot,2);
spl_ind = [375, 962, 1590, 2501, 3556, 4888, 6681, 8939, 11488, new_new_length];

% plot data in 2D
figure
set(gcf,'RendererMode','manual','Renderer','painters')
for i = 1:numsma
    if i == 1
        plot(matrix_new_new_tot(1:spl_ind(1),1),matrix_new_new_tot(1:spl_ind(1),3), ...
            ... 'dk','MarkerSize',3,'MarkerFaceColor','k')
        hold on
    else
        plot(matrix_new_new_tot(spl_ind(i-1)+1:spl_ind(i),1), ...
            ... matrix_new_new_tot(spl_ind(i-1)+1:spl_ind(i),3), 'd','MarkerSize',3, ...
            ... 'Color',[i*0.08 i*0.08 i*0.08],'MarkerFaceColor',[i*0.08 i*0.08 i*0.08])
        hold on
    end
end
set(gca,'FontName','Palatino','FontSize',12)
legend('h=200km','h=250km','h=300km','h=350km','h=400km','h=450km','h=500km','h=550km', ...
    ... 'h=600km','h=650km','Location','EastOutside')
xlabel('Dauer [min]')
ylabel('erzeugte Energie [Wh]')
ylim([11 21])
grid on
hold off

```

### F.3 Skript zur Evolution der Bahnhöhe

Das hier wiedergegebene Skript dient der Erzeugung der Rohdaten für die Abschätzung der Evolution der Bahnhöhe für zukünftige CLAVIS-Missionen für den Fall hoher Sonnenaktivität (siehe Abschnitt 10.7). Dabei wird hier eine Simulation über das Jahr 1991 durchgeführt. Die Skripte für die Fälle mittlerer und niedriger Sonnenaktivität unterscheiden sich lediglich durch den simulierten Zeitraum und finden sich auf der CD zur Arbeit.

#### evolution\_of\_altitude\_1991.m

```

function [out] = evolution_of_altitude_1991()

%-----
display('Hook to running STK...')
%-----

uiApp = actxGetRunningServer('STK9.application');
root = uiApp.Personality2;

```

```

% define units
root.UnitPreferences.SetCurrentUnit('DateFormat', 'JDate');
root.UnitPreferences.SetCurrentUnit('DistanceUnit', 'm');
root.UnitPreferences.SetCurrentUnit('TimeUnit', 'sec');
root.UnitPreferences.SetCurrentUnit('AngleUnit', 'deg');

%-----
display('Create new objects and set their properties...')
%-----
% create new scenario, define scenario object
root.NewScenario('CLAVIS_Matlab_neu');
Scenario = root.GetObjectFromPath('*/');
jd_start = jday(1991,01,01,00,00,00);
jd_stop = jday(1992,01,01,00,00,00);
Scenario.StartTime = jd_start;
Scenario.StopTime = jd_stop;

% create Satellite CLAVIS
Satellite = Scenario.Children.New('eSatellite', 'CLAVIS');
Satellite.SetPropagatorType('ePropagatorHPOP');
Satellite.Propagator.StartTime = jd_start;
Satellite.Propagator.StopTime = jd_stop;
Satellite.Propagator.Step = 60;
Satellite.Propagator.ForceModel.SolarRadiationPressure.SRPMModel.Model.Cr = 1.21;
Satellite.Propagator.ForceModel.SolarRadiationPressure.SRPMModel.Model.AreaMassRatio = 0.0104;
Satellite.Propagator.ForceModel.SolarRadiationPressure.UseBoundaryMitigation = 1;
Satellite.Propagator.ForceModel.Drag.AreaMassRatio = 0.0104;
Satellite.Propagator.ForceModel.Drag.SetSolarFluxGeoMagType('eSolarFluxGeoMagUseFile');
Satellite.Propagator.ForceModel.Drag.SolarFluxGeoMag.File = ...
    ... 'C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\AGI\STK 9\DynamicEarthData\ ...
    ... stkFluxGeoMag.fxm';
Satellite.Propagator.ForceModel.MoreOptions.RadiationPressure.IncludeAlbedo = 1;
Satellite.Propagator.ForceModel.MoreOptions.RadiationPressure.IncludeThermal = 1;
Satellite.Propagator.ForceModel.MoreOptions.RadiationPressure.Ck = 1.21;
% hier anderes Flächen- zu Massenverhältnis! keine Reflektorplatte!
Satellite.Propagator.ForceModel.MoreOptions.RadiationPressure.AreaMassRatio = 0.0104;
Satellite.Propagator.Integrator.DoNotPropagateBelowAlt = 90000;

% semimajor axis
a = 6578140:50000:7028140;
le = (jd_stop-jd_start)*1440+1;
alt = zeros(length(a),le);

for i = 1:length(a)
    Satellite.Propagator.InitialState.Representation.AssignClassical ...
        ... ('eCoordinateSystemJ2000', a(i), 0.0, 45, 0.0, 0.0, 45);
    Satellite.Propagator.Propagate();
    DataProvider = Satellite.DataProviders.Item('Classical Elements').Group.Item('J2000');
    Results = DataProvider.ExecElements(jd_start, jd_stop, 60, {'Semi-major Axis'; 'Time'});
    Data = Results.DataSets;
    Data_num = Data.GetDataSetByName('Semi-major Axis').GetValues;
    for j = 1:length(Data_num)
        alt(i,j) = (Data_num{j}-6378140)/1000;
        if alt(i,j) < 0
            alt(i,j) = 0;
        end
    end
    if i == length(a)
        time_num = Data.GetDataSetByName('Time').GetValues;
    end
    progress = i*100/length(a)
end

% convert time vector cell -> numeric
time = zeros(length(time_num),1);
for i = 1:length(time_num)
    time(i) = time_num{i}-jd_start;
end

% save data
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\evolution_of_altitude\ ...
    ... 1991_high_sol\time_high_sol.mat', 'time');
save('G:\12_SR\30_SAS\Klemich\Simulationsergebnisse\evolution_of_altitude\ ...
    ... 1991_high_sol\altitude_high_sol.mat', 'alt');

% close scenario
Scenario.Unload;

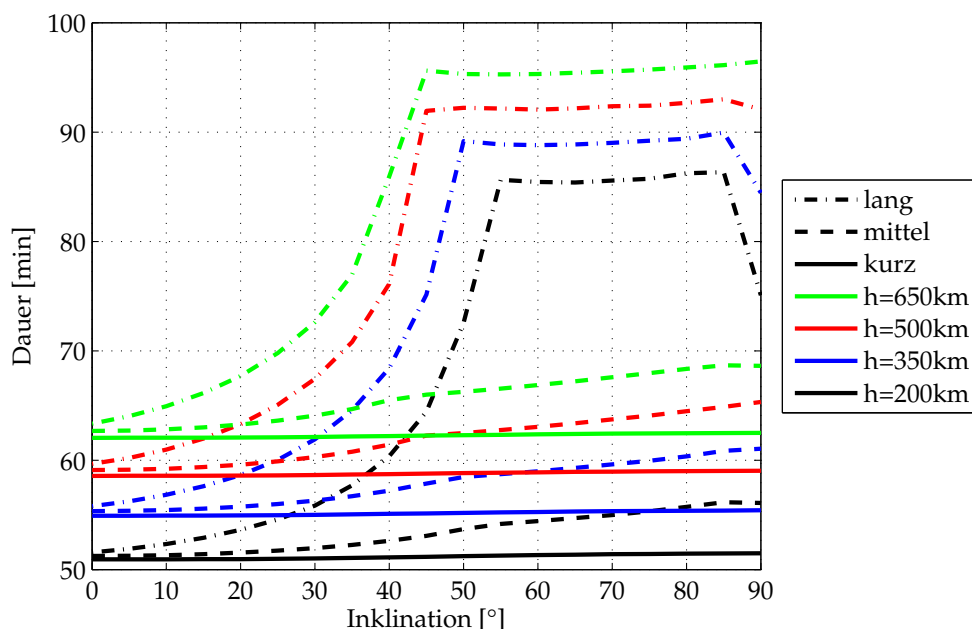
out = 1;
end

```

# Anhang G

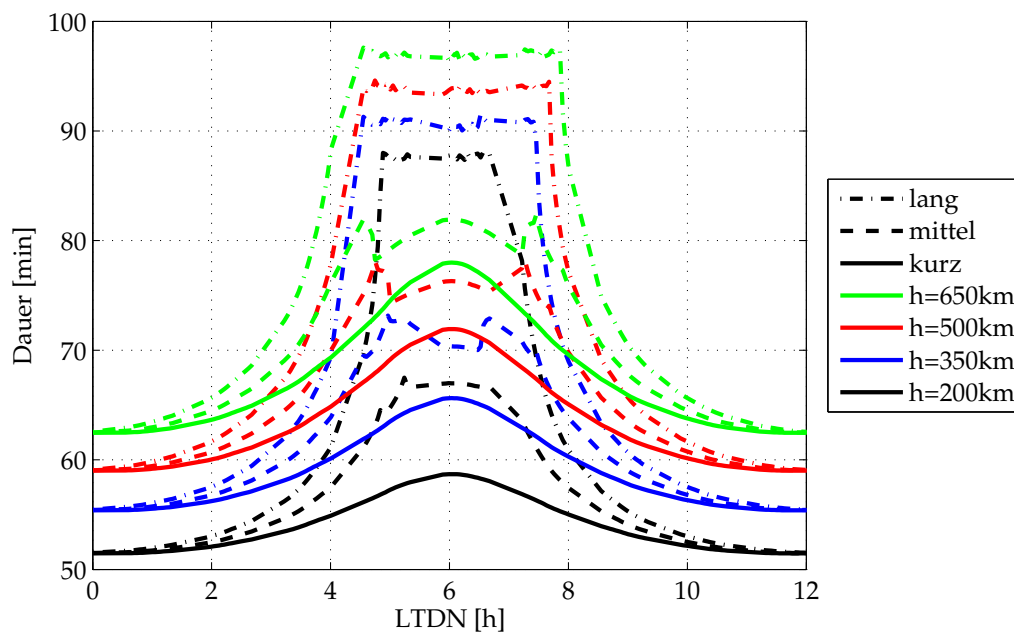
## Zweidimensionale Diagramme

In diesem Abschnitt werden die dreidimensionalen Diagramme aus Kapitel 10 erneut in zwei Dimensionen dargestellt, damit Werte besser abgelesen werden können. Dabei werden stets Kurvenscharen dargestellt, deren Parameter die Bahnhöhe ist und die anderen relevanten Variablen (Dauer, Energie, LTDN, Inklination) werden auf den Achsen aufgetragen. Die verschiedenen Bahnhöhen werden dabei mit verschiedenfarbigen Verläufen dargestellt und ggf. vorhandene verschiedene Fälle (besten, schlechtesten, mittleren) mit verschiedenen Linienarten. Um die Übersichtlichkeit der Diagramme nicht zu beeinträchtigen wird nur eine reduzierte Anzahl an Bahnhöhen berücksichtigt.

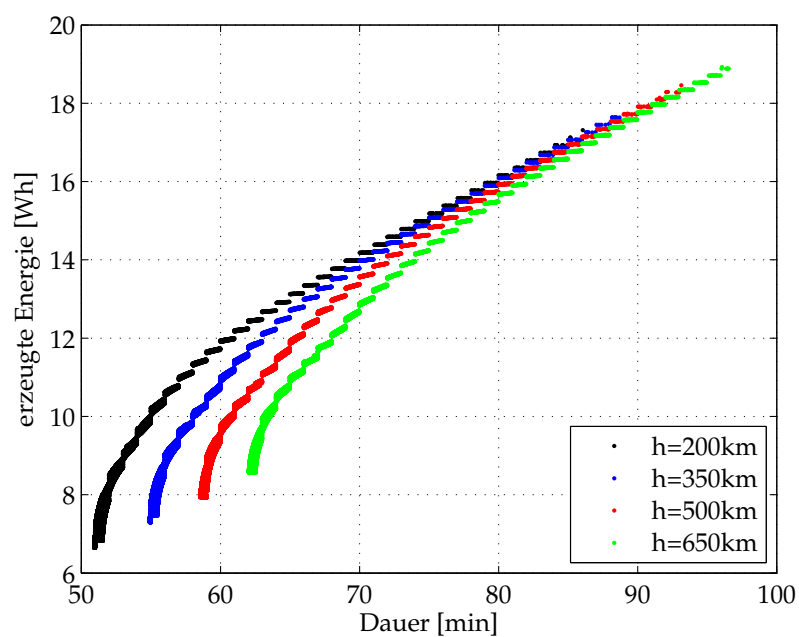


**Abbildung G.1:** Verläufe von kürzester, mittlerer und längster Sonnenlichtdauer als Funktion der Inklination für verschiedene Bahnhöhen, Mittelwerte über alle Startzeitpunkte und -rektaszensionen, zu Abbildung 10.2

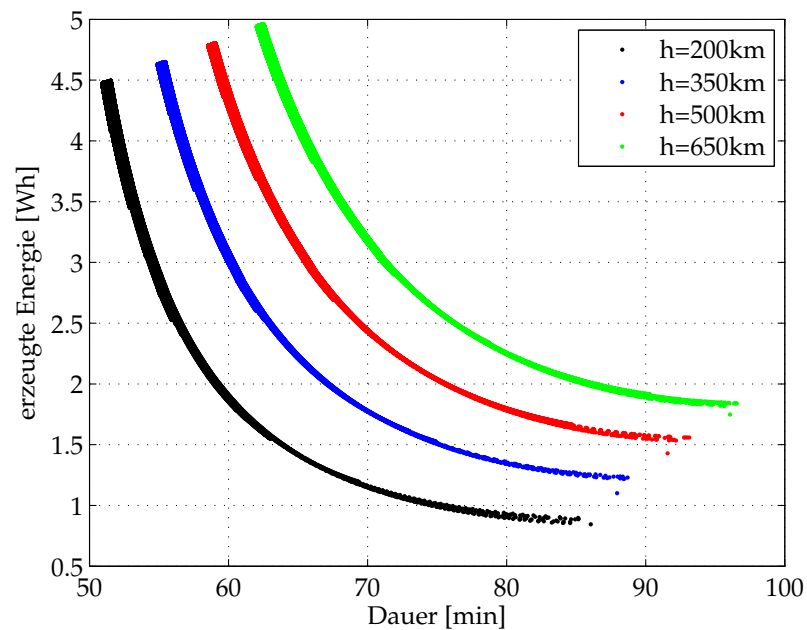




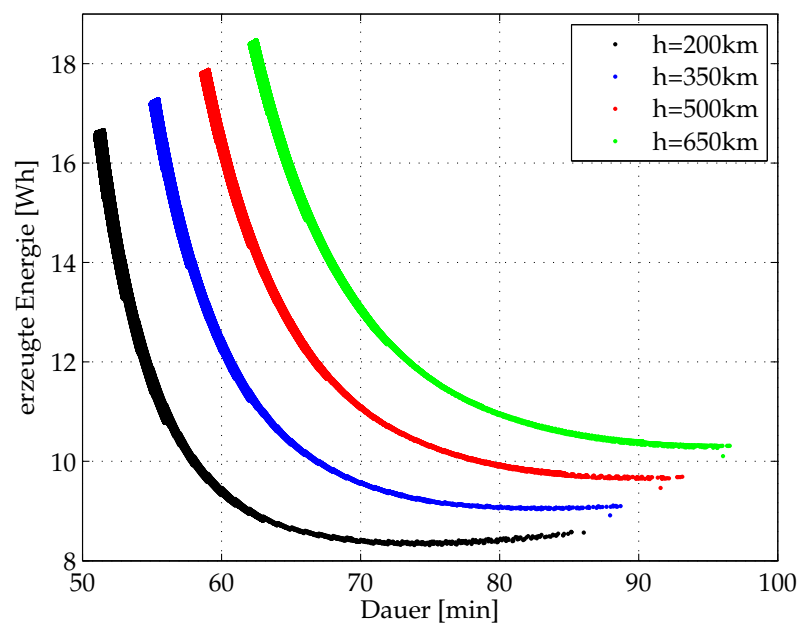
**Abbildung G.2:** Verläufe von kürzester, mittlerer und längster Sonnenlichtdauer für SSOs als Funktion der LTDN für verschiedene Bahnhöhen, zu den Abbildungen 10.4 und 10.5



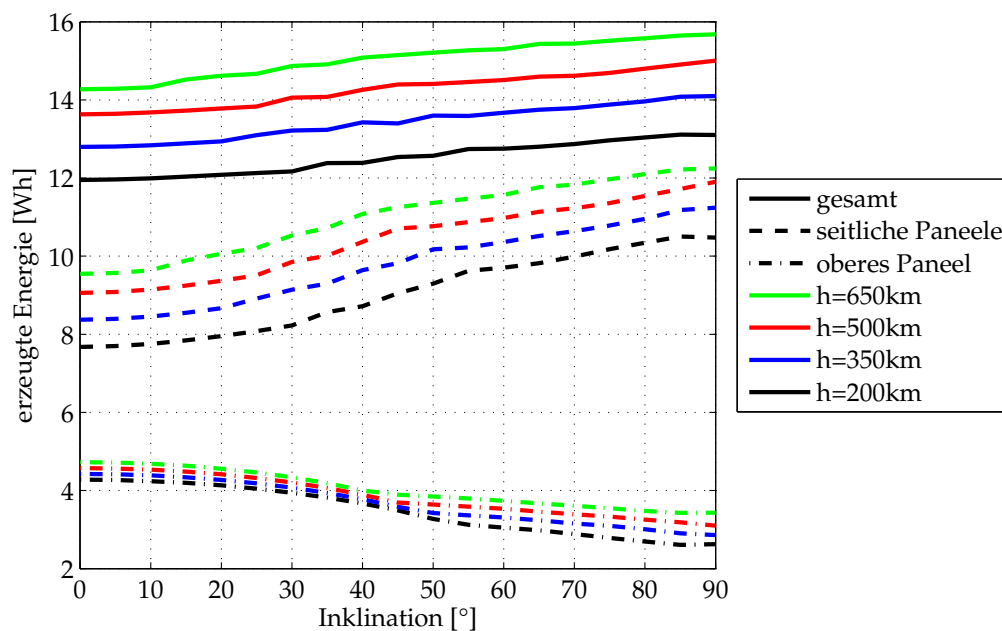
**Abbildung G.3:** Verläufe der erzeugten Energie aller seitlichen Paneele der Standardkonfiguration als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen, zu Abbildung 10.7



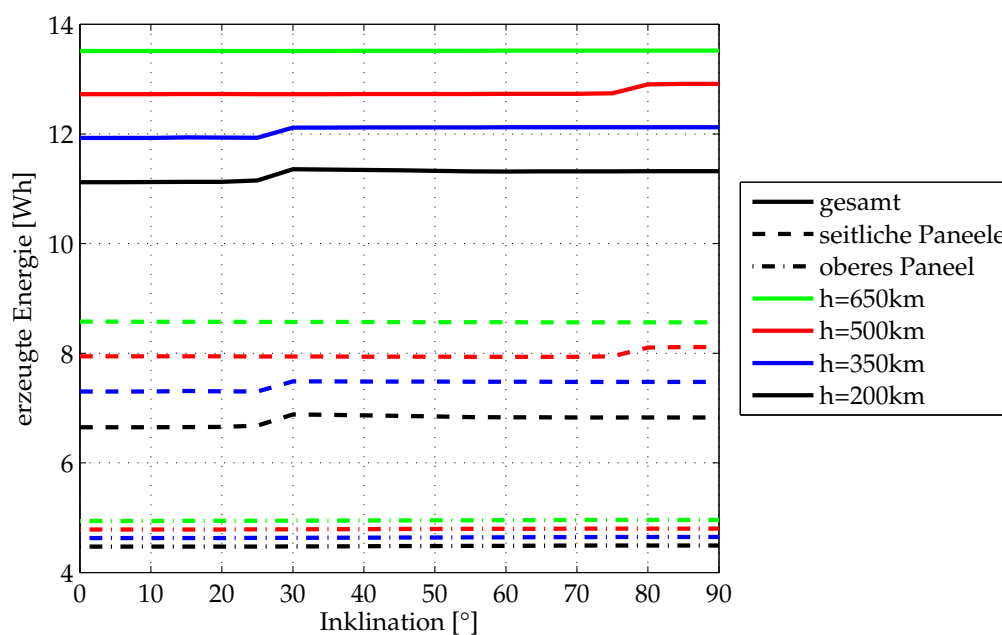
**Abbildung G.4:** Verlauf der erzeugten Energie des oberen Paneels als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen, zu Abbildung 10.8



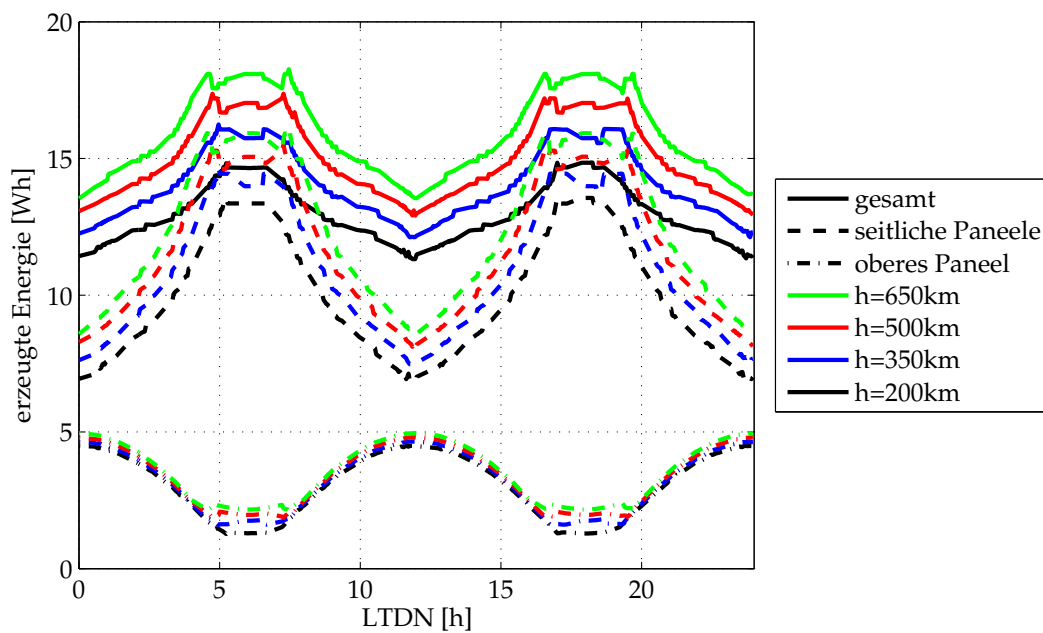
**Abbildung G.5:** Verlauf der erzeugten Energie aller seitlichen Paneele der Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen als Funktion der Dauer der Sonnenlichtperiode für verschiedene Bahnhöhen, zu Abbildung 10.10



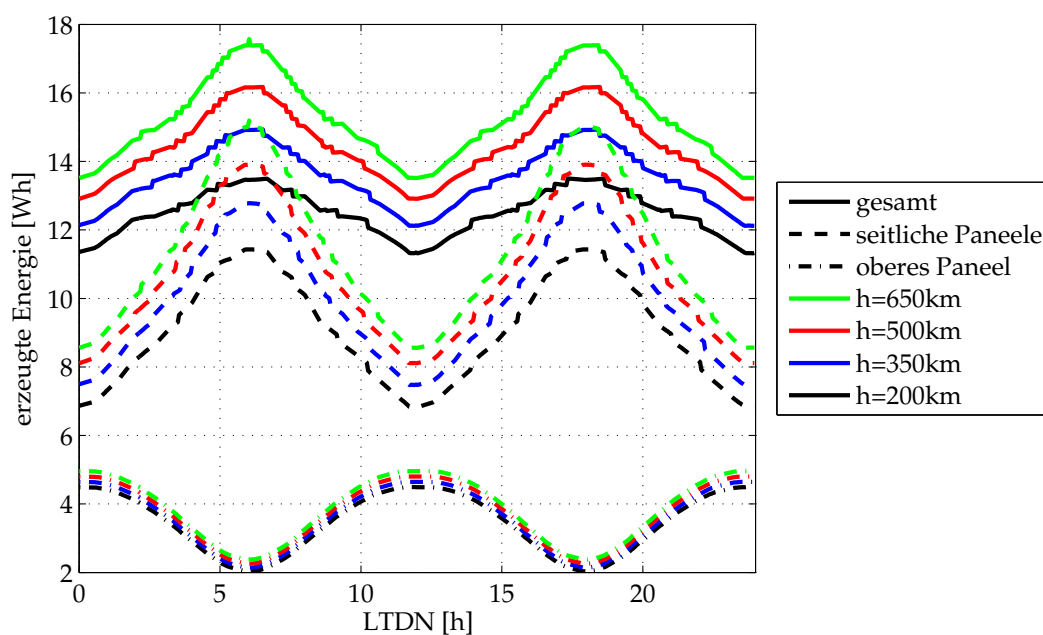
**Abbildung G.6:** Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.12



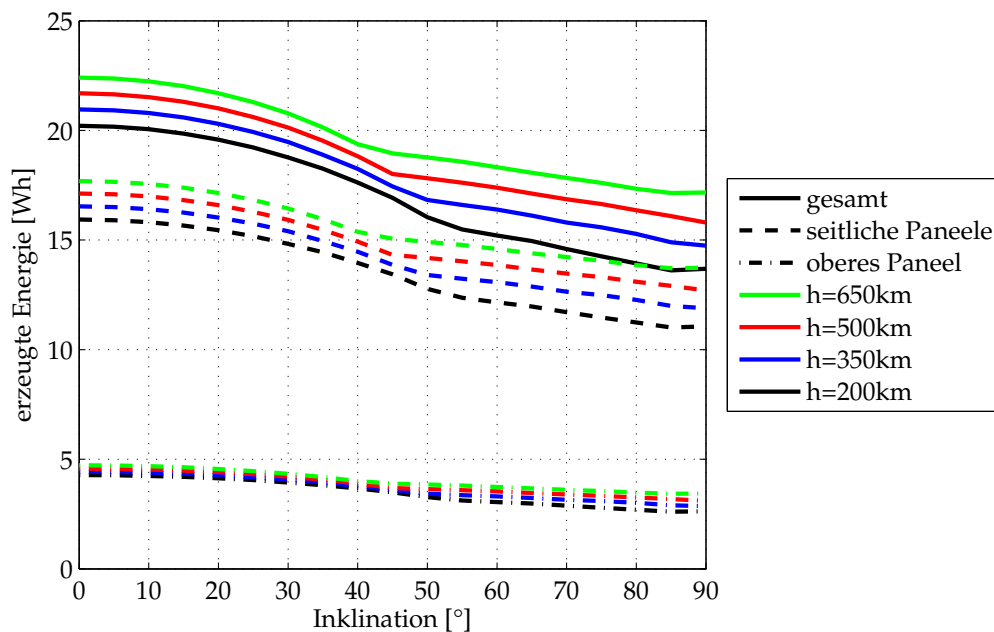
**Abbildung G.7:** Minimalwerte der durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugten Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.13



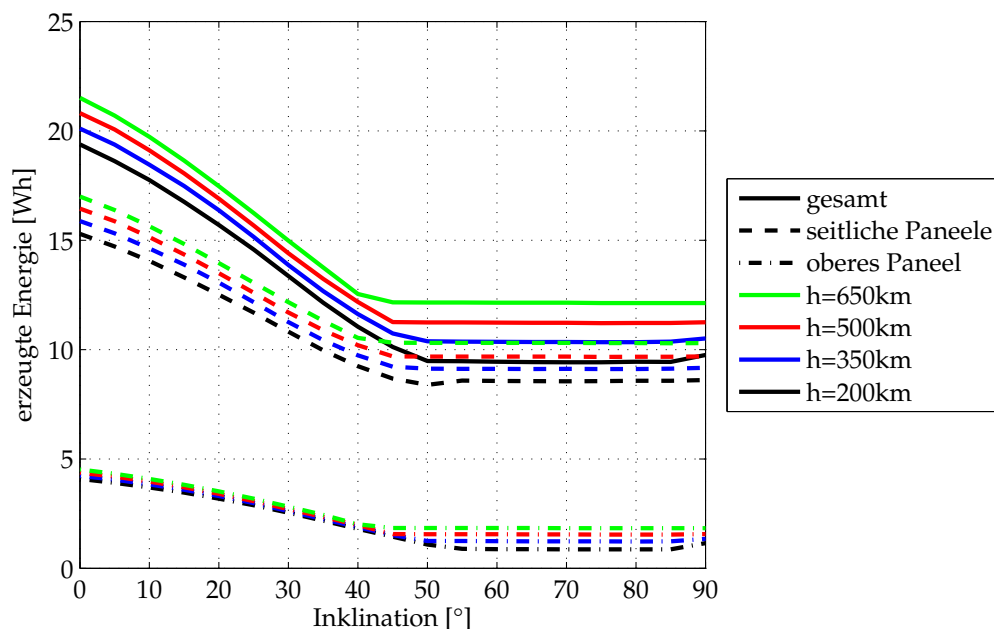
**Abbildung G.8:** Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der LTDN für verschiedene Bahnhöhen, zu den Abbildungen 10.14 und 10.15



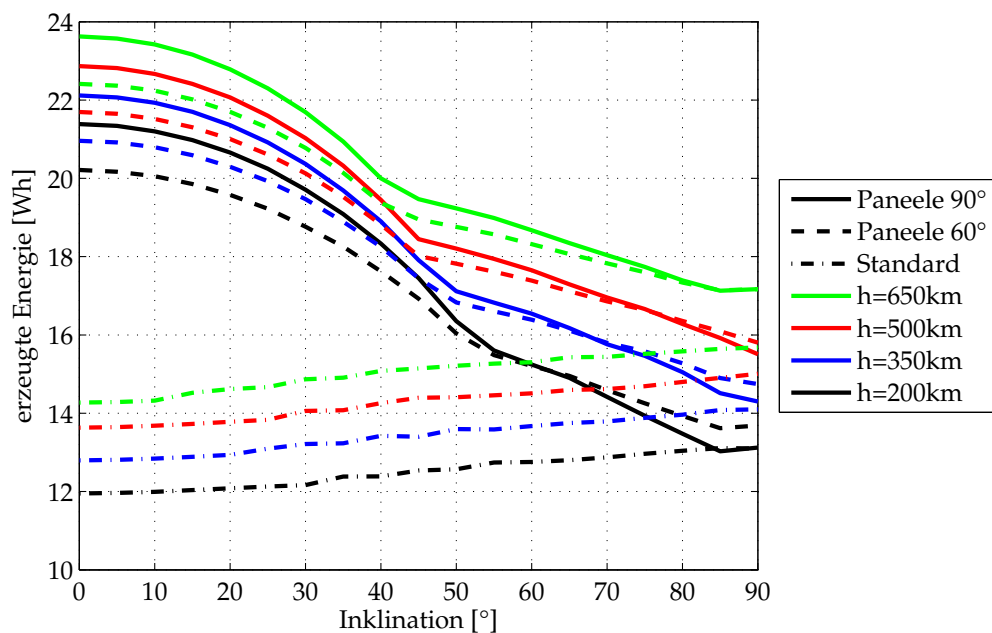
**Abbildung G.9:** Minimale durch das obere Paneel, die seitlichen Paneele und die Standardkonfiguration insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.16



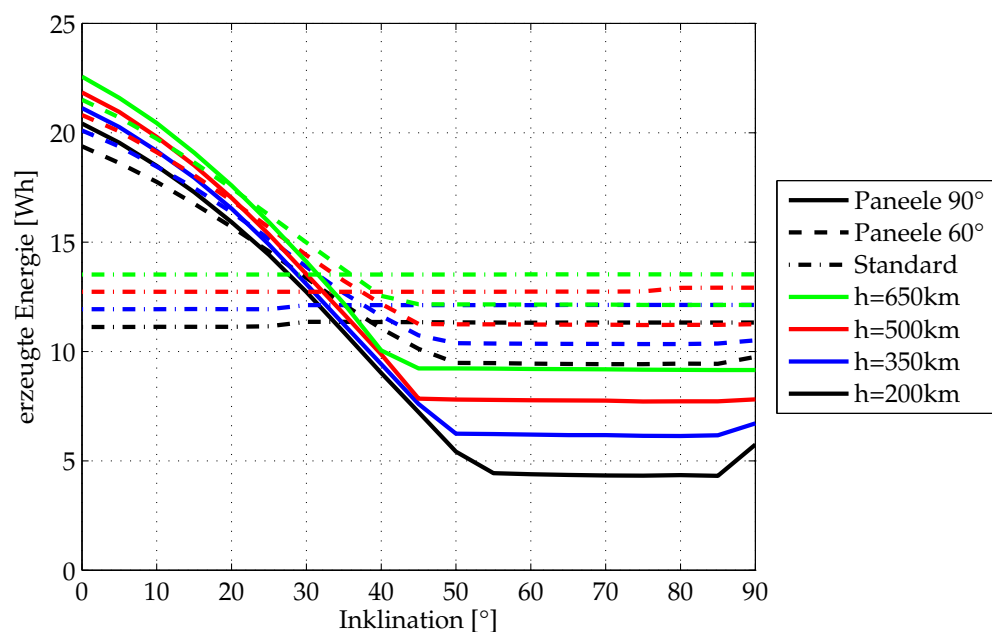
**Abbildung G.10:** Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um 60° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.17



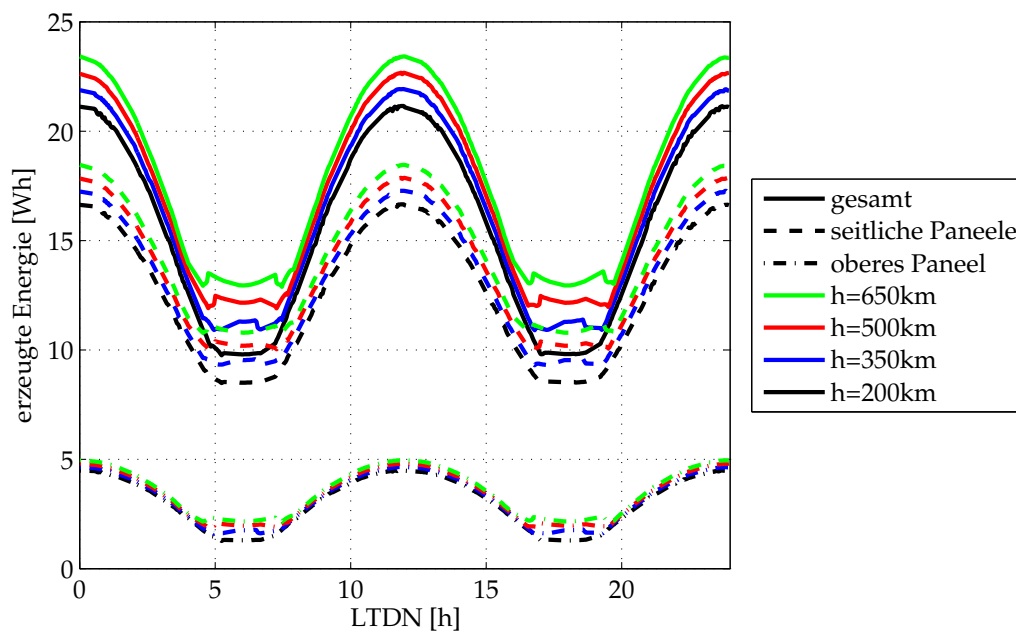
**Abbildung G.11:** Durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um 60° ausgeklappten Paneelen insgesamt erzeugte Energie bei minimaler Gesamtenergie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.18



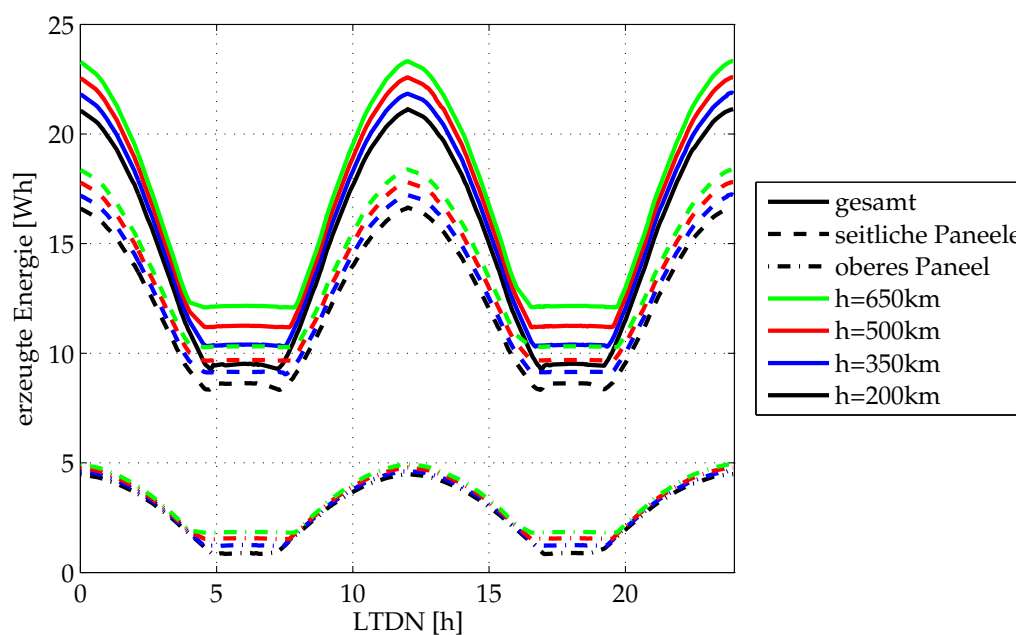
**Abbildung G.12:** Durchschnittlich durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.19



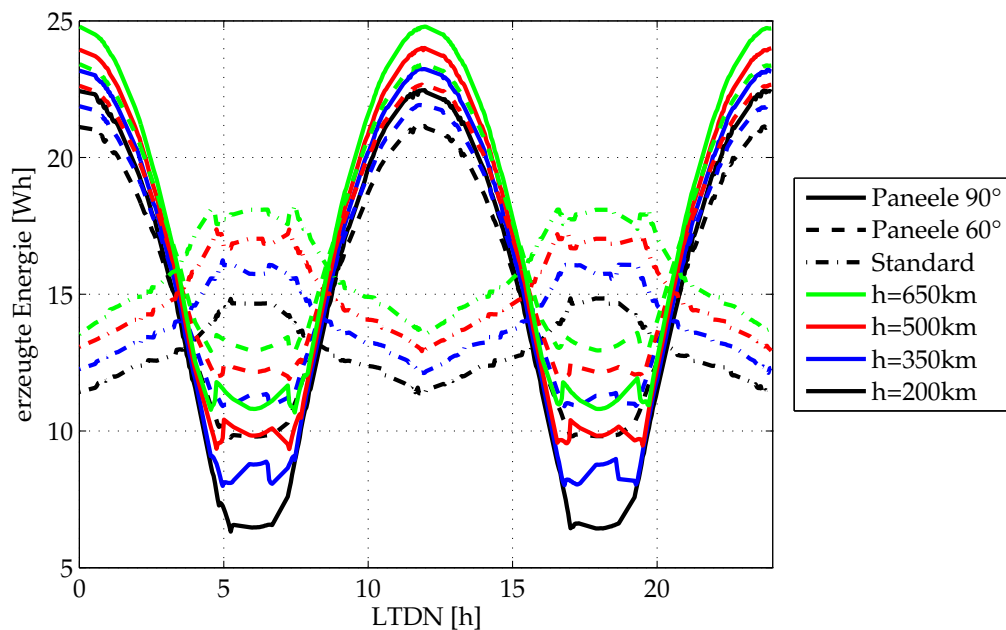
**Abbildung G.13:** Minimal durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.20



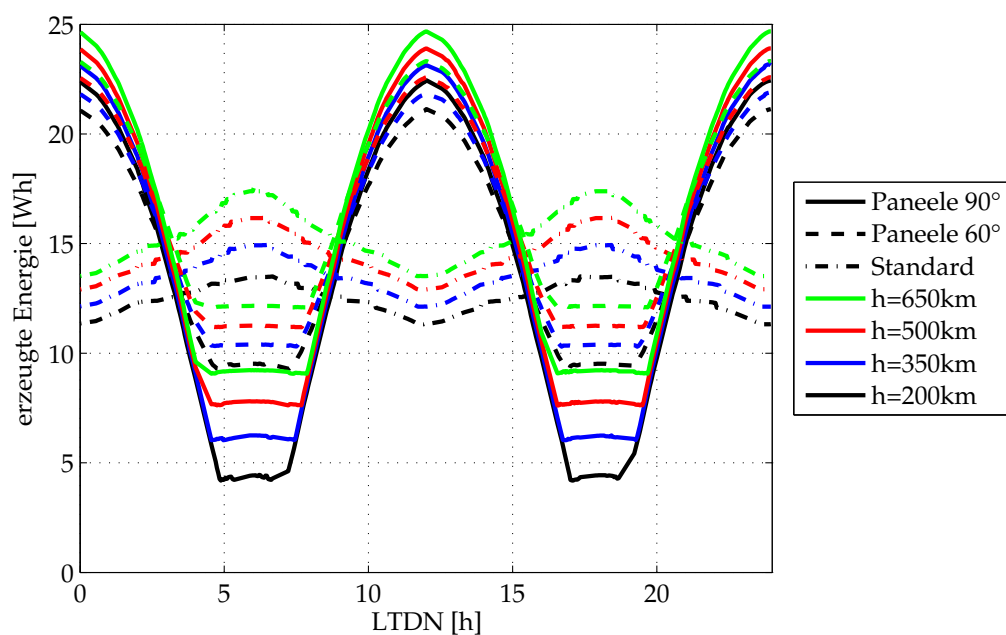
**Abbildung G.14:** Durchschnittlich durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen, zu Abbildung 10.21



**Abbildung G.15:** Im schlechtesten Fall durch das obere Paneel, die seitlichen Paneele und die Konfiguration mit um  $60^\circ$  ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der Inklination für verschiedene Bahnhöhen



**Abbildung G.16:** Durchschnittlich durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der LTDN für SSOs für verschiedene Bahnhöhen, zu Abbildung 10.22

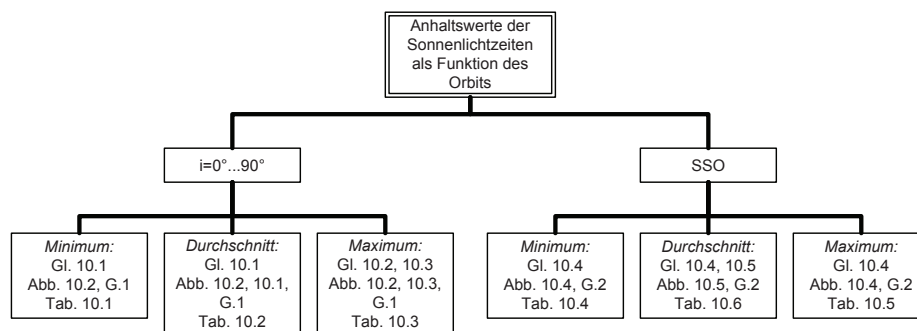


**Abbildung G.17:** Minimal durch die Standardkonfiguration und die Konfigurationen mit um 60° bzw. 90° ausgeklappten Paneelen insgesamt erzeugte Energie über eine Mission von einem Jahr in Abhängigkeit der LTDN für SSOs für verschiedene Bahnhöhen

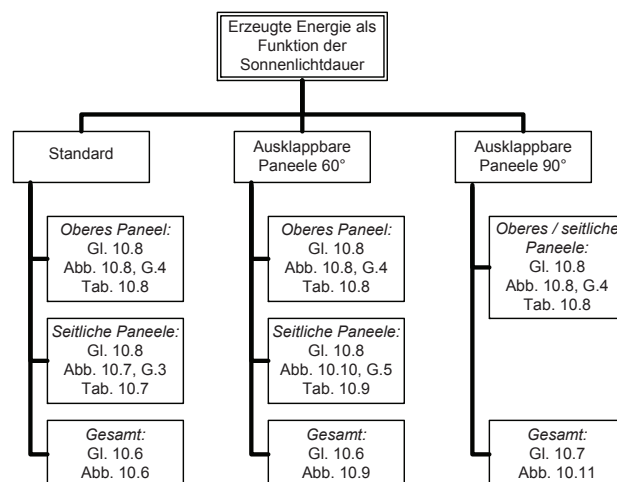


# Anhang H

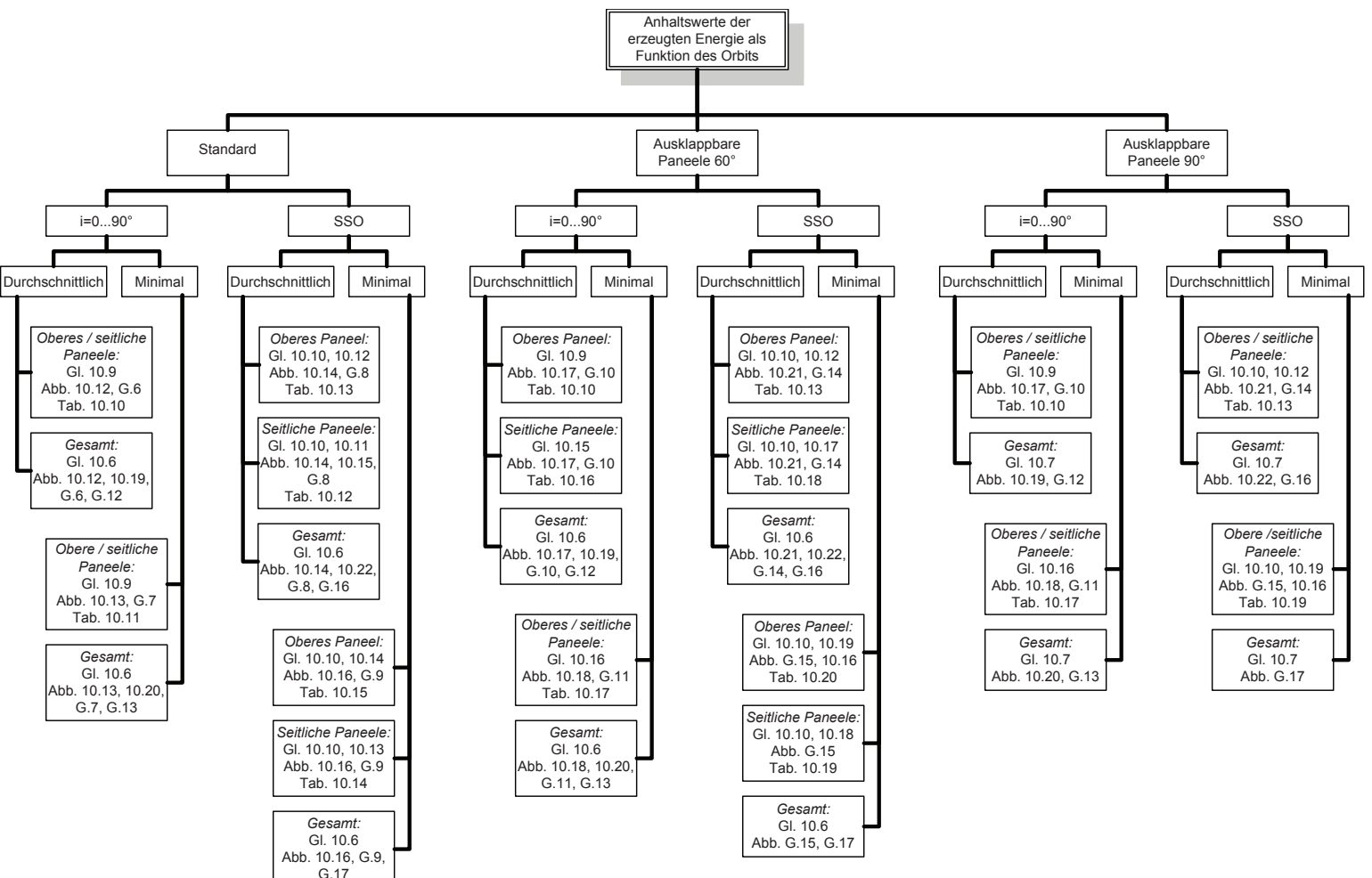
## Übersicht über die zu verwendenden Formeln, Diagramme und Tabellen



**Abbildung H.1:** Übersicht über die zur Abschätzung der kürzesten, längsten und durchschnittlichen Sonnenlichtzeiten aus den Orbitparametern zu verwendenden Gleichungen (Gl.), Abbildungen (Abb.) und Tabellen (Tab.)



**Abbildung H.2:** Übersicht über die zur Abschätzung der erzeugten Energie aus der Dauer der Sonnenlichtzeiten zu verwendenden Gleichungen (Gl.), Abbildungen (Abb.) und Tabellen (Tab.)



**Abbildung H.3:** Übersicht über die zur Abschätzung der erzeugten Energie aus den Orbitparametern zu verwendenden Gleichungen (Gl.), Abbildungen (Abb.) und Tabellen (Tab.)

# Anhang I

## Projektplanung

### I.1 Work Breakdown Structure

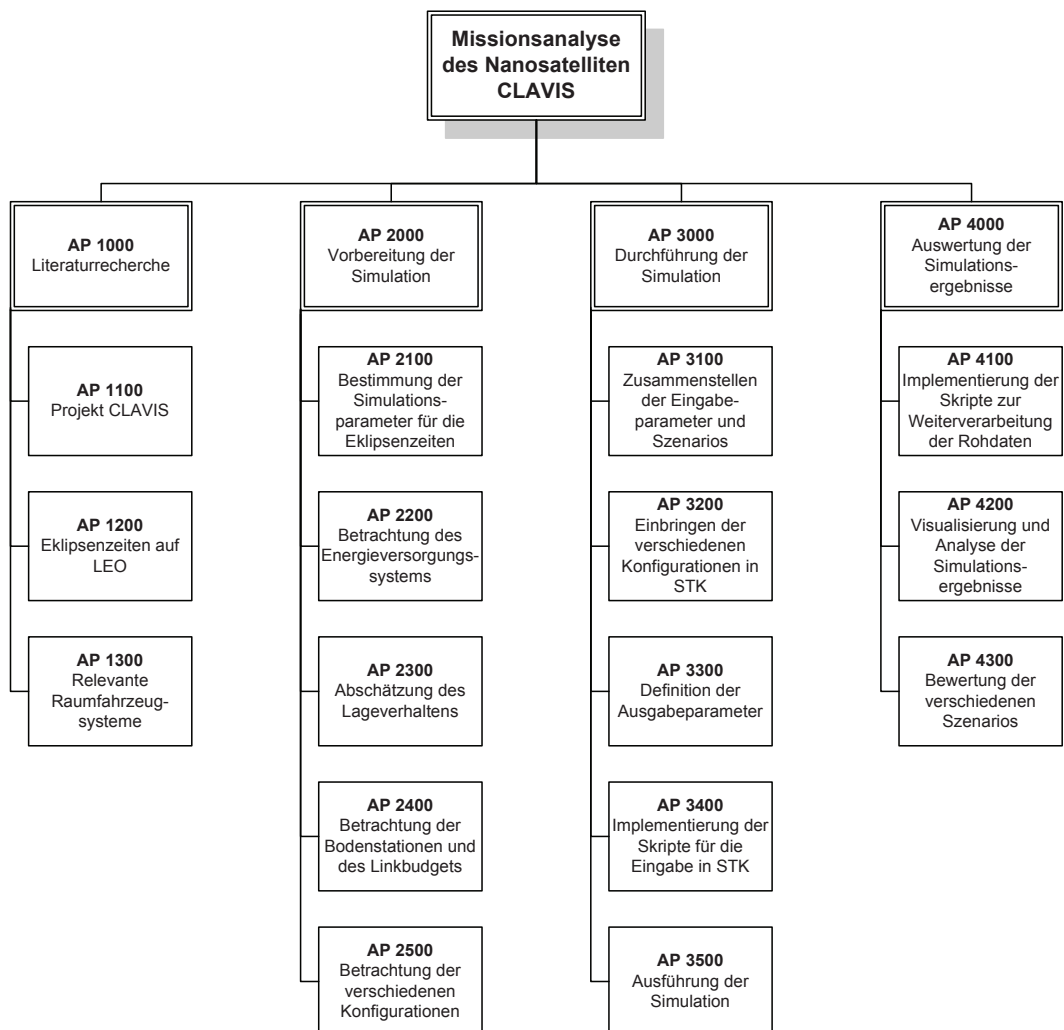


Abbildung I.1: Work Breakdown Structure

## I.2 Work Package Description

Missionsanalyse für den Nanosatelliten CLAVIS		AP 1100
Titel	Projekt CLAVIS	Seite: 1 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub>	
Ende	T <sub>0</sub> + 3 Tage	Dauer: 3 Tage
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>• Überblick über das Projekt CLAVIS</li><li>• Kenntnisse der Geometrie, der relevanten Subsysteme und der möglichen Orbits</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>• institutsinterne Dokumentation zum Projekt</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>• AP 2000 Vorbereitung der Simulation</li><li>• AP 3000 Durchführung der Simulation</li><li>• AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>• Erlangen von ausreichenden Kenntnissen des Satellitensystems, um geeignete Simulationen durchführen zu können</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>• Kenntnisse der Subsysteme von CLAVIS und Verständnis der Missionsziele</li><li>• Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 1200
Titel	Eklipsenzeiten auf LEO	Seite: 2 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 3 Tage	
Ende	T <sub>0</sub> + 1 Woche	Dauer: 2 Tage
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>• Verständnis der Entstehung von Eklipsen in erdnahen Orbits</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>• Literatur zur Geometrie von Raumfahrtmissionen</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>• AP 2100 Bestimmung der Simulationsparameter für die Eklipsenzeiten</li><li>• AP 2200 Betrachtung des Energieversorgungssystems</li><li>• AP 3000 Durchführung der Simulation</li><li>• AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>• Erlangen von ausreichenden Kenntnissen zur Geometrie von Eklipsen auf LEO, um geeignete Simulationen durchführen zu können</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>• Kenntnisse der Einflussfaktoren und Verständnis der Geometrie von Eklipsen</li><li>• Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 1300
Titel	Relevante Raumfahrzeugsysteme	Seite: 3 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 1 Woche	
Ende	T <sub>0</sub> + 2 Wochen	Dauer: 1 Woche
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>• Kenntnisse der Funktionsweise und der Berechnung der für die Betrachtung relevanten Raumfahrzeugsysteme</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>• Literatur zum Design von Raumfahrzeugen und Raumfahrtmissionen</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>• AP 2200 Betrachtung des Energieversorgungssystems</li><li>• AP 2300 Abschätzung des Lageverhaltens</li><li>• AP 2400 Betrachtung der Bodenstationen und des Linkbudgets</li><li>• AP 3000 Durchführung der Simulation</li><li>• AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>• Erlangen von Kenntnissen zum Energieversorgungssystem (Solarpaneele, Batterien), Lageregelungssystem (Gravitationsgradientenstabilisierung, Magnet Torquer) und Kommunikationssystem</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>• Verständnis der in CLAVIS vorhandenen Komponenten der relevanten Subsysteme</li><li>• Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 2100
Titel	Bestimmung der Simulationsparameter für die Eklipsenzeiten	Seite: 4 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 2 Wochen	
Ende	T <sub>0</sub> + 3 Wochen	Dauer: 1 Woche
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>Bestimmung der Ein- und Ausgabeparameter für die Simulation, so dass alle Einflussfaktoren auf die Eklipsenzeiten berücksichtigt werden</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>AP 1200 Eklipsenzeiten auf LEO</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>AP 3100 Zusammenstellen der Eingabeparameter und Szenarios</li><li>AP 3400 Implementierung der Skripte für die Eingabe in STK</li><li>AP 3500 Ausführung der Simulation</li><li>AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>Betrachtung der Einflüsse der Bahnhöhe, der Inklination, der RAAN, der J<sub>2</sub>-Störungen, der jahreszeitlichen Änderungen sowie der Geometrie der Erdbahn auf die Eklipsenzeiten</li><li>Verständnis der Anwendung vereinfachender analytischer Gleichungen zur Berechnung der Eklipsenzeiten</li><li>Übersicht über die Einflussfaktoren zur Bestimmung geeigneter Simulationszeitpunkte und –räume sowie geeigneter Bahndaten</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>Liste von Ein- und Ausgabeparametern für die Simulation</li><li>Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 2200
Titel	Betrachtung des Energieversorgungssystems	Seite: 5 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 3 Wochen	
Ende	T <sub>0</sub> + 5 Wochen	Dauer: 2 Wochen
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>Abschätzung der durch die Solarpaneele erzeugten Energie und der Ladezyklen der Batterie</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>AP 1100 Projekt CLAVIS</li><li>AP 1300 Relevante Raumfahrzeugsysteme</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>AP 3100 Zusammenstellen der Eingabeparameter und Szenarios</li><li>AP 3300 Definition der Ausgabeparameter</li><li>AP 3400 Implementierung der Skripte für die Eingabe in STK</li><li>AP 3500 Ausführung der Simulation</li><li>AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>Verständnis der verschiedenen Verluste beim Betrieb der Solarzellen mit Hilfe von Angaben des Herstellers und der Literatur</li><li>Verständnis der Kennwerte des Lade- und Entladeverhaltens der Batterie</li><li>Zusammentragen des Energieverbrauchs der verschiedenen Komponenten unter verschiedenen Bedingungen</li><li>Aufstellen eines Energiebudgets unter Berücksichtigung verschiedener Betriebsmodi des Satelliten und den zuvor zusammengetragenen Kennwerten zur späteren Abschätzung, ob die erzeugte Energie ausreicht</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>Zahlenwerte für die zur Verfügung stehende Energie bei gegebenen Einflussparametern unter Berücksichtigung von Verlusten</li><li>Beitrag zum technischen Report</li></ul>		



Missionsanalyse für den Nanosatelliten CLAVIS		AP 2300
Titel	Abschätzung des Lageverhaltens	Seite: 6 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 5 Wochen	
Ende	T <sub>0</sub> + 6 Wochen	Dauer: 1 Woche
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>Abschätzung des zeitlichen Verlaufs der Lage des Satelliten im Raum unter Berücksichtigung der Eigenschaften des vorhandenen Lageregelungssystems</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>AP 1100 Projekt CLAVIS</li><li>AP 1300 Relevante Raumfahrzeugsysteme</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>AP 3100 Zusammenstellen der Eingabeparameter und Szenarios</li><li>AP 3300 Definition der Ausgabeparameter</li><li>AP 3400 Implementierung der Skripte für die Eingabe in STK</li><li>AP 3500 Ausführung der Simulation</li><li>AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>Verständnis des Lageverhaltens von CLAVIS auf Grundlage des vorhandenen Lageregelungssystems</li><li>Verständnis von Störeinflüssen auf die Lageregelung und die daraus resultierenden Änderungen der Lage</li><li>Bestimmung von Verläufen der Lage des Satelliten im Raum in geeigneter Form (Quaternionen, Eulerwinkel) für verschiedene Szenarios</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>geeignete Werte zur Beschreibung des Lageverhaltens für die Eingabe in STK</li><li>Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 2400
Titel	Betrachtung der Bodenstationen und des Linkbudgets	Seite: 7 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 6 Wochen	
Ende	T <sub>0</sub> + 8 Wochen	Dauer: 2 Wochen
Bearbeiter	Kai-Sören Klemich	

**Ziele:**

- Abschätzung der möglichen Datenübertragungsraten der vorhandenen Kommunikationssysteme zu den Bodenstationen

**Input:**

- AP 1100 Projekt CLAVIS
- AP 1300 Relevante Raumfahrzeugsysteme

**Schnittstellen zu anderen APs:**

- AP 3100 Zusammenstellen der Eingabeparameter und Szenarios
- AP 3300 Definition der Ausgabeparameter
- AP 3400 Implementierung der Skripte für die Eingabe in STK
- AP 3500 Ausführung der Simulation
- AP 4000 Auswertung der Simulationsergebnisse

**Zweck:**

- Verständnis der Geometrie von Funkverbindungen und möglicher Bodenkontaktzeiten auf LEO, ggf. Definition von Bodenstationsnetzwerken
- Verständnis der vereinfachten Berechnung der Übertragungsraten für das Kommunikationssystem auf CLAVIS
- Verständnis der möglichen Verluste bei der Datenübertragung und deren Berechnung
- Zusammentragen der für die Bestimmung der möglichen Datenübertragungsraten nötigen Parameter zur Eingabe in STK

**Ergebnisse:**

- Ein- und Ausgabeparameter für die Simulation zur Abschätzung des Linkbudgets
- ggf. Listen mit Bodenstationsnetzwerken
- Beitrag zum technischen Report

Missionsanalyse für den Nanosatelliten CLAVIS		AP 2500
Titel	Betrachtung der verschiedenen Konfigurationen	Seite: 8 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 8 Wochen	
Ende	T <sub>0</sub> + 9 Wochen	Dauer: 1 Woche
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>Analyse der verschiedenen Konfigurationen von CLAVIS im Hinblick auf mögliche Abschattungseffekte und zusätzliche Payloads</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>AP 1100 Projekt CLAVIS</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>AP 3200 Einbringen der verschiedenen Konfigurationen in STK</li><li>AP 3500 Ausführung der Simulation</li><li>AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>Definition einer Standardkonfiguration von CLAVIS unter Berücksichtigung des vorläufigen Designs, evtl. mit fehlenden Paneelen für potentielle Nutzlasten</li><li>Definition der AISat-Konfiguration von CLAVIS unter Berücksichtigung des endgültigen Designs und der Komponenten, die eine Abschattung bewirken können</li><li>Definition einer Konfigurationen mit entfaltbaren Paneelen unter Berücksichtigung des vorläufigen Designs und eventueller Abschattungseffekte</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>Listen der relevanten Komponenten der verschiedenen Konfigurationen mit ihren Abmessungen im Hinblick auf Abschattungseffekte</li><li>Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 3100
Titel	Zusammenstellen der Eingabeparameter und Szenarios	Seite: 9 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 9 Wochen	
Ende	T <sub>0</sub> + 11 Wochen	Dauer: 2 Wochen
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>• Zusammenführen der für die einzelnen Subsysteme relevanten Eingabeparameter für die Verarbeitung mit STK</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>• AP 2000 Vorbereitung der Simulation</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>• AP 3300 Definition der Ausgabeparameter</li><li>• AP 3400 Implementierung der Skripte für die Eingabe in STK</li><li>• AP 3500 Ausführung der Simulation</li><li>• AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>• Einarbeitung in die Software STK</li><li>• Zusammenstellen aller in AP 2000 recherchierten bzw. berechneten Parameter, Szenarios und Konfigurationen für die Simulationen</li><li>• Einteilung der verschiedenen zu simulierenden Szenarios in sinnvolle Simulationsläufe, so dass die Simulation möglichst effektiv durchgeführt werden kann</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>• Listen mit den verschiedenen auf STK abgestimmten Simulationsparametern und den nötigen Simulationsläufen</li><li>• Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 3200
Titel	Einbringen der verschiedenen Konfigurationen in STK	Seite: 10 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 11 Wochen	
Ende	T <sub>0</sub> + 12 Wochen	Dauer: 1 Woche
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>• Einbringen der verschiedenen Konfigurationen von CLAVIS in STK als *.mdl-Dateien</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>• AP 2500 Betrachtung der verschiedenen Konfigurationen</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>• AP 3500 Ausführung der Simulation</li><li>• AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>• Einarbeiten in die Definition von Geometrien und Solarpaneelen in STK</li><li>• Erstellen der Geometriedefinitionen der verschiedenen Konfigurationen in STK zur Vorbereitung der Simulation</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>• *.mdl-Dateien für alle Konfigurationen von CLAVIS</li><li>• Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 3300
Titel	Definition der Ausgabeparameter	Seite: 11 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 12 Wochen	
Ende	T <sub>0</sub> + 14 Wochen	Dauer: 2 Wochen
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>Definition aller Ausgabeparameter, die zur kompletten Auswertung der Simulationsergebnisse notwendig sind</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>AP 2000 Vorbereitung der Simulation</li><li>AP 3100 Zusammenstellen der Eingabeparameter und Szenarios</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>AP 3400 Implementierung der Skripte für die Eingabe in STK</li><li>AP 3500 Ausführung der Simulation</li><li>AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>Zusammenstellen der Ausgabeparameter, die nötig sind, um alle gewünschten Ergebnisse daraus bestimmen zu können (Eklipsenzeiten, erzeugte Energie, übertragbare Datenraten)</li><li>Definition der später zu erzeugenden Diagramme mit den entsprechenden Achsen</li><li>Übersicht über alle notwendigen Ausgabeparameter der verschiedenen Simulationsläufe</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>Liste mit auf STK abgestimmten Ausgabeparametern für die Simulation</li><li>Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 3400
Titel	Implementierung der Skripte für die Eingabe in STK	Seite: 12 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 14 Wochen	
Ende	T <sub>0</sub> + 18 Wochen	Dauer: 4 Wochen
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>• Implementierung aller notwendigen Skripte für die Simulation unter Berücksichtigung aller Ein- und Ausgabeparameter in Matlab</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>• AP 2000 Vorbereitung der Simulation</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>• AP 3500 Ausführung der Simulation</li><li>• AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>• Einarbeitung in die Schnittstelle zwischen STK und Matlab</li><li>• Definition von Funktionen und ausführbaren Skripten zur Übersichtlichkeit der verwendeten Skripte und Dateien</li><li>• Definition der Speicherung der gewonnenen Rohdaten zur Weiterverarbeitung</li><li>• Implementierung der Skripte für die verschiedenen Simulationsläufe unter Berücksichtigung der notwendigen Ein- und Ausgabeparameter</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>• Matlab-Skripte, mit denen die gewünschten Simulationsergebnisse erzielt und geeignet abgespeichert werden</li><li>• Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 3500
Titel	Ausführung der Simulation	Seite: 13 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 18 Wochen	
Ende	T <sub>0</sub> + 21 Wochen	Dauer: 3 Wochen
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>Durchführung der eigentlichen Simulationsläufe mit STK</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>AP 2000 Vorbereitung der Simulation</li><li>AP 3100 Zusammenstellen der Eingabeparameter und Szenarios</li><li>AP 3200 Einbringen der verschiedenen Konfigurationen in STK</li><li>AP 3300 Definition der Ausgabeparameter</li><li>AP 3400 Implementierung der Skripte für die Eingabe in STK</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>AP 4000 Auswertung der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>Test der Skripte mit verringerter Genauigkeit zur Verringerung der Rechenzeit und ggf. Implementierung von Verbesserungen</li><li>Durchführung der Simulationsläufe mit der endgültigen Genauigkeit</li><li>Bereitstellung der Rohdaten aus den Simulationen zur Auswertung</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>in geeigneter Weise abgespeicherte Ausgaben von STK</li><li>Beitrag zum technischen Report</li></ul>		



Missionsanalyse für den Nanosatelliten CLAVIS		AP 4100
Titel	Implementierung der Skripte zur Weiterverarbeitung der Rohdaten	Seite: 14 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 21 Wochen	
Ende	T <sub>0</sub> + 23 Wochen	Dauer: 2 Wochen
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>• Implementierung aller notwendigen Skripte zur Analyse der aus der Simulation erhaltenen Ausgaben in Matlab</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>• AP 2000 Vorbereitung der Simulation</li><li>• AP 3000 Durchführung der Simulation</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>• AP 4200 Visualisierung und Analyse der Simulationsergebnisse</li><li>• AP 3300 Bewertung der verschiedenen Szenarios</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>• Definition von Funktionen und ausführbaren Skripten zur Übersichtlichkeit der verwendeten Skripte und Dateien</li><li>• Implementierung der Skripte zur Berechnung der gewünschten Endergebnisse (maximale, minimale und durchschnittliche Eklipsenzeiten, zur Verfügung stehende Energie, übertragbare Datenraten) aus den Rohdaten</li><li>• Test der Skripte</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>• Matlab-Skripte zur Auswertung der aus den Simulationen gewonnenen Rohdaten</li><li>• Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 4200
Titel	Visualisierung und Analyse der Simulationsergebnisse	Seite: 15 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 23 Wochen	
Ende	T <sub>0</sub> + 25 Wochen	Dauer: 2 Wochen
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>• Darstellung aller gewünschten Simulationsergebnisse in Diagrammen und zusammenfassenden Beschreibungen</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>• AP 2000 Vorbereitung der Simulation</li><li>• AP 3000 Durchführung der Simulation</li><li>• AP 4100 Implementierung der Skripte zur Weiterverarbeitung der Rohdaten</li></ul>		
<b>Schnittstellen zu anderen APs:</b> <ul style="list-style-type: none"><li>• AP 4300 Bewertung der verschiedenen Szenarios</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>• Implementierung von Skripten zur Darstellung der zuvor berechneten Ergebnisse in den gewünschten Diagrammen</li><li>• Implementierung von Skripten zur Bestimmung von Näherungsgleichungen für bestimmte Verläufe</li><li>• Beschreibung und Auswertung der gezeigten Diagramme</li><li>• anschauliche, zusammenfassende Darstellung der Simulationsergebnisse in grafischer und schriftlicher Form</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>• Diagramme und ggf. Näherungsgleichungen für bestimmte Verläufe</li><li>• Zusammenfassung der Ergebnisse der Simulation</li><li>• Beitrag zum technischen Report</li></ul>		

Missionsanalyse für den Nanosatelliten CLAVIS		AP 4300
Titel	Bewertung der verschiedenen Szenarios	Seite: 16 von 16
Verantwortlicher	Kai-Sören Klemich	Version: 1.2
		Datum: 17.11.2010
Beginn	T <sub>0</sub> + 25 Wochen	
Ende	T <sub>0</sub> + 26 Wochen	Dauer: 1 Woche
Bearbeiter	Kai-Sören Klemich	
<b>Ziele:</b> <ul style="list-style-type: none"><li>Bewertung aller betrachteten Orbits, Konfigurationen, Szenarios etc.</li></ul>		
<b>Input:</b> <ul style="list-style-type: none"><li>AP 2000 Vorbereitung der Simulation</li><li>AP 3000 Durchführung der Simulation</li><li>AP 4100 Implementierung der Skripte zur Weiterverarbeitung der Rohdaten</li><li>AP 4200 Visualisierung und Analyse der Simulationsergebnisse</li></ul>		
<b>Zweck:</b> <ul style="list-style-type: none"><li>abschließende Bewertung der Simulationsergebnisse zur Identifikation von besonders geeigneten und besonders ungeeigneten Szenarios im Hinblick auf Eklipsenzeiten, zur Verfügung stehender Energie und übertragbare Datenmengen anhand der Diagramme und Näherungsgleichungen</li></ul>		
<b>Ergebnisse:</b> <ul style="list-style-type: none"><li>Zusammenfassende Bewertung der Simulationsergebnisse unter Berücksichtigung der verschiedenen Eingabeparameter</li><li>Beitrag zum technischen Report</li></ul>		

## I.3 Zeitplanung

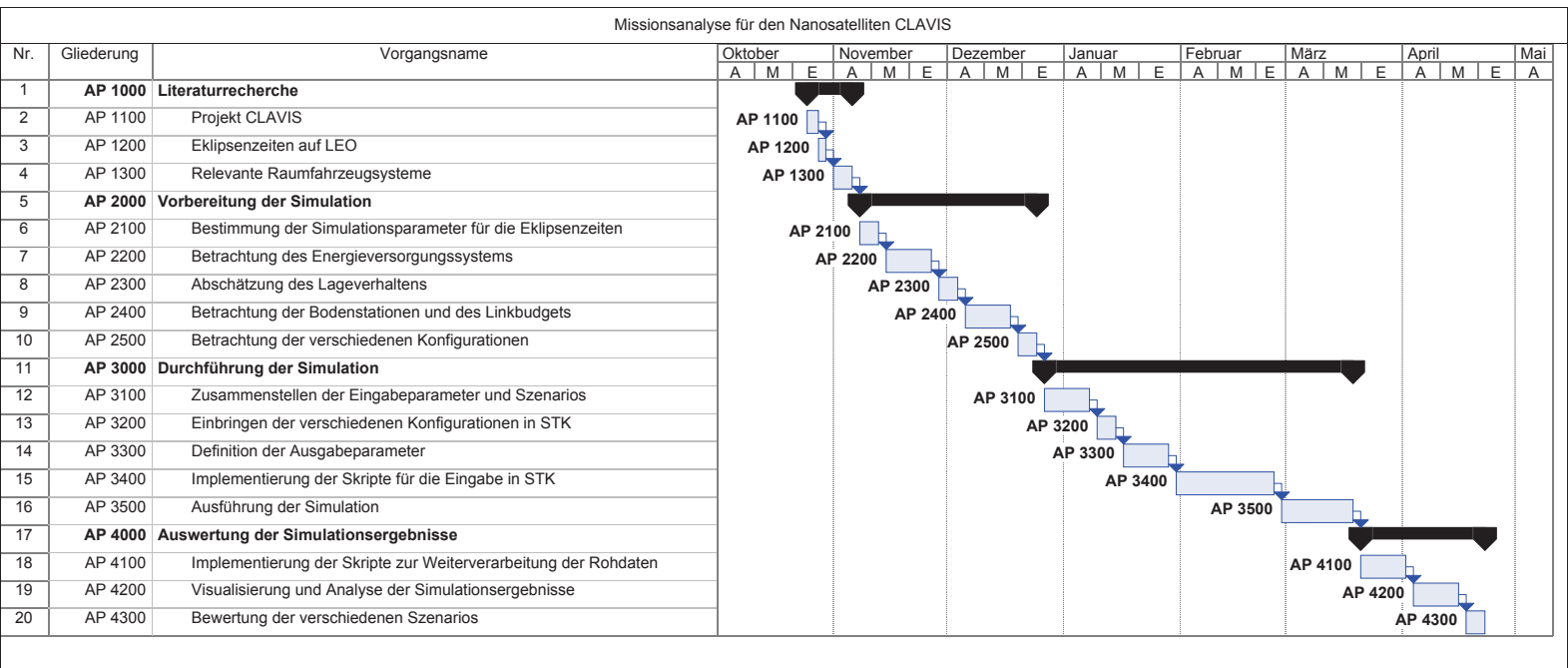


Abbildung I.2: Projektplanung als Gantt-Diagramm